Universidad de Granada

Algoritmos Evolutivos para la extracción de Reglas de Asociación Cuantitativas

Tesis Doctoral

Diana Martín Rodríguez

Granada, Junio de 2014

Departamento de Ciencias de la Computación
e Inteligencia Artificial

Editor: Editorial de la Universidad de Granada
Autor: Diana Martín Rodríguez
D.L.: GR 2103-2014
ISBN: 978-84-9083-131-1

Universidad de Granada

Algoritmos Evolutivos para la extracción de Reglas de Asociación Cuantitativas

MEMORIA QUE PRESENTA

Diana Martín Rodríguez

PARA OPTAR AL GRADO DE DOCTOR EN INFORMÁTICA

Junio de 2014

DIRECTORES

Jesús Alcalá Fernández
Francisco Herrera Triguero

Departamento de Ciencias de la Computación
e Inteligencia Artificial

La doctoranda Diana Martín Rodríguez y los directores de la tesis los doctores D. Francisco Herrera Triguero y D. Jesús Alcalá Fernández garantizamos, al firmar esta tesis doctoral, que el trabajo ha sido realizado por la doctoranda bajo la dirección de los directores de la tesis y hasta donde nuestro conocimiento alcanza, en la realización del trabajo, se han respetado los derechos de otros autores a ser citados, cuando se han utilizado sus resultados o publicaciones.

Granada, Mayo de 2014

La Doctoranda

Fdo: Diana Martín Rodríguez

El Director

Fdo: D. Jesús Alcalá Fernández

El Director

Fdo: D. Francisco Herrera Triguero

Agradecimientos

Dedico este proyecto a mi familia: a mis padres, de los que estoy muy orgullosa, por todo lo que me han enseñado, por su apoyo incondicional y constante, por sus consejos y su madurez. Hoy les devuelvo el fruto de lo que sembraron. A mis hermanas por sus consejos y por estar ahí siempre que las necesito. Y en especial a mi esposo, por dedicarme su tiempo, por tranquilizarme en los momentos de mayor estrés y por compartir esta alegría conmigo. A mis amigas, por acompañarme con entusiasmo a lo largo de esta tarea.

A mis directores Francisco Herrera y Jesús Alcalá, por creer en mí y darme toda la confianza de que juntos podíamos lograrlo. Les agradezco haberme guiado, empujado hacia delante y hasta las interminables correcciones que tanto me enseñaron.

A Alejandro Rosete, por haber sembrado en mí la semilla de la investigación, por guiarme y por su ayuda en las diferentes ocasiones.

A mis compañeros de departamento de la CUJAE, con los que siempre he podido contar.

A mis compañeros y amigos de Granada por toda su ayuda y por haberme hecho sentir como en casa.

Como dijera Robert L. Stevenson : "*No midas el éxito por la cosecha de hoy. Mide el éxito por las semillas que plantas hoy* ".

Por eso hoy doy las gracias a todos aquellos que dedicaron su tiempo y su energía en aportar el granito de arena tan necesario para que todo saliera como esperaba.

GRACIAS, GRACIAS, GRACIAS

Resumen

El contenido de esta tesis trata sobre los algoritmos evolutivos para la obtención de reglas de asociación cuantitativas positivas y negativas con mejores propiedades de calidad y diversidad en el conjunto de reglas obtenido. En una fase inicial se estudia el diseño de algoritmos multi-objetivos para la extracción de reglas de asociación que integre distintas medidas de calidad. En segundo lugar se aborda el problema de obtener un conjunto de reglas de asociación de alta diversidad apoyándonos en los algoritmos genéticos basados en nichos.

Los objetivos llevados a cabo fueron:

1. Diseñar una variante evolutiva multi-objetivo para la extracción de reglas de asociación cuantitativas positivas que integre distintas medidas de calidad de las reglas.

2. Diseñar un enfoque evolutivo multi-objetivo para extraer un conjunto reducido de reglas de asociación cuantitativas positivas y negativas de buena calidad y con un alto cubrimiento de la base de datos.

3. Mejorar la diversidad de las reglas obtenidas mediante el diseño de un algoritmo genético basado en nichos.

Para llevar a cabo estos objetivos se han desarrollado diferentes algoritmos evolutivos para obtener reglas de asociación con una alta calidad y diversidad en el conjunto de reglas obtenido.

Primero, se presenta el método QAR-CIP-NSGA-II, un nuevo modelo evolutivo multi-objetivo para extraer reglas de asociación con un buen equilibrio entre las diferentes medidas de interés y el cubrimiento de la base de datos. Esta propuesta permite obtener conjuntos de reglas muy específicas con valores muy

altos para las medidas de interés, proporcionando al usuario reglas de muy alta calidad. Por otro lado, se ha propuesto MOPNAR, un nuevo modelo evolutivo basado en el algoritmo evolutivo multi-objetivo MOEA/D-DE, que resulta muy útil para obtener conjuntos más reducidos de reglas a partir de extraer reglas de asociación positivas y negativas, que aportan información interesante sobre toda la base de datos. MOPNAR alcanza los mejores valores de cubrimiento de las base de datos. Por último, se ha presentado NIGAR, un nuevo algoritmo genético basado en nichos para obtener un conjunto muy diverso de reglas de asociación con buena calidad y cubrimiento de la BD, evitando la obtención de reglas redundantes que proporcionen información similar sobre la base de datos. Destacar que los tres métodos evolutivos permiten obtener reglas de asociación interesantes, por lo que su uso dependerá de las necesidades específicas de cada usuario. Los usuarios podrán basar su selección en las potencialidades que cada uno ofrece.

Índice

Índice de figuras

Índice de tablas

Tabla de Acrónimos

AEMO	—	Algoritmo Evolutivo Multi-Objetivo	23
AGN	—	Algoritmo Genético basado en Nichos	26
BD	—	Base de Datos	7
PE	—	Población Externa	7
RAC	—	Reglas de Asociación Cuantitativa	7
RACPN	—	Reglas de Asociación Cuantitativa Positiva y Negativa	16

Introducción

A Planteamiento

En la última década, la revolución digital ha proporcionado medios relativamente económicos y accesibles de recolección y almacenamiento de datos. Este crecimiento ilimitado de datos ha provocado que el proceso de extracción de conocimiento sea más difícil y, en la mayoría de los casos, conduzca a problemas de escalabilidad y/o la complejidad [RBV03].

La minería de datos pretende resolver este problema al extraer conocimiento interesante a partir de conjuntos de datos grandes y complejos. Una de las técnicas de minería de datos más utilizada para extraer conocimiento interesante a partir de bases de datos ha sido el descubrimiento de reglas de asociación [HK06].

Las reglas de asociación son utilizadas para representar e identificar dependencias entre los atributos de una base de datos [ZZ02]. Estas reglas son expresiones del tipo $X \rightarrow Y$, donde X y Y son conjuntos de ítems (parejas atributo-valor) y cumplen que $X \cap Y = \oslash$. Esto significa que si todos los ítems de X están en un ejemplo de la base de datos entonces todos los ítems de Y están también en el ejemplo con una alta probabilidad, y X y Y no tienen ningún ítem en común [AIS93, AS94]. El uso de reglas de asociación para resolver problemas del mundo real ha sido una práctica muy extendida en muchas áreas, como la biología [CC09], la medicina [CRLA11], etc.

Muchos estudios previos para extraer reglas de asociación han sido enfocados sobre base de datos con valores binarios o discretos, sin embargo los datos en las aplicaciones del mundo real normalmente están compuestos por valores cuantitativos. Debido a ello, varios estudios han sido presentados para extraer reglas de asociación cuantitativas a partir de bases de datos con valores cuantitativos, donde normalmente cada ítem es un par *atributo-intervalo* [SA96]. Muchos de

estos métodos se centran solo en extraer reglas positivas sin poner atención a las reglas negativas que pueden ser interesantes al expresar, por ejemplo, la ausencia de Y ante la presencia de X ($X \rightarrow \neg Y$). Las reglas de asociación negativas pueden incluir ítems negativos en el antecedente, en el consecuente o en ambos. Debido a ello, durante los últimos años algunos investigadores han propuesto métodos para extraer reglas de asociación cuantitativas positivas y negativas [AA06, WZZ04].

Los algoritmos clásicos difícilmente pueden extraer reglas de asociación a partir de base de datos cuantitativas debido a que los atributos numéricos normalmente contienen muchos valores distintos. Para evitar este problema un método normalmente utilizado es particionar el dominio de los atributos numéricos en intervalos. El soporte de un valor determinado es probablemente bajo, mientras que el soporte de un intervalo es mucho mayor. De esta forma, podríamos extraer reglas de asociación cuantitativas, sin embargo, la partición de los datos puede ser un problema crítico en la extracción de este tipo de reglas porque la información no está clasificada. Además, dichos intervalos pueden tener una gran influencia sobre las reglas obtenidas.

En los últimos años muchos investigadores han propuesto algoritmos evolutivos [ES03] para extraer reglas de asociación cuantitativas [AFFPBH10] como una variante a los algoritmos de obtención de reglas de asociación clásicos, los cuales suelen extraer una enorme cantidad de reglas al realizar una exploración exhaustiva del espacio de búsqueda con un consecuente problema de escalabilidad. Además los algoritmos evolutivos, particularmente los algoritmos genéticos [Gol89] son considerados una de las técnicas de búsqueda más exitosas para problemas complejos y han demostrado ser muy buenos en el aprendizaje y la extracción de conocimiento. Al utilizarse los algoritmos genéticos se debe definir la función objetivo a optimizar. Para evaluar las reglas de asociación se suele trabajar con dos medidas para conocer su calidad: cobertura y confianza. La cobertura (también denominada soporte) de una regla indica el número de casos que cubre la regla y la confianza (también llamada precisión) mide el número de casos en que se cumple la regla cuando se cumplen las premisas.

Normalmente, los algoritmos genéticos propuestos para extraer reglas de asociación definen su función objetivo teniendo en cuenta solo las medidas mencionadas anteriormente. Sin embargo, varios autores han señalado algunos inconvenientes de estas medidas que conduce a encontrar reglas de baja calidad [BBSV02]. La confianza no detecta independencia estadística o dependencia negativa entre los ítems debido a que no tiene en cuenta el soporte del consecuente. Asimismo, los ítems con soporte muy alto pueden generar reglas de baja calidad porque cual-

quier conjunto de ítems parece ser un buen predictor de ellos. Por esta razón, los algoritmos genéticos propuestos miden la calidad de las reglas a través de medidas que no describen completamente la calidad del conocimiento descubierto.

La necesidad de incluir nuevas medidas de calidad de las reglas de asociación implica la transformación del problema de extracción de reglas de asociación hacia un problema multi-objetivo [AA08, GN04] (en lugar de un simple objetivo), donde se traten de optimizar explícitamente varias funciones objetivos a la vez, ofreciendo al decisor final un conjunto de soluciones no dominadas de mayor calidad en función de varias medidas.

Los algoritmos evolutivos multi-objetivos [Deb01, CLV02] son un mecanismo interesante para tratar problemas de naturaleza multi-objetivo, ya que generan una familia de soluciones igualmente válidas, en las que cada solución tiende a satisfacer un criterio en mayor medida que otro. Por esta razón, algunos algoritmos evolutivos multi-objetivos han sido aplicados para extraer reglas de asociación cuantitativas (considerando varias medidas como objetivos) [GN04, AA08], donde cada solución en el frente de Pareto representa una regla de asociación diferente con un grado de equilibrio entre las diferentes medidas.

La calidad de las reglas obtenidas a través de los algoritmos genéticos también puede verse afectada por la similitud que tengan las reglas entre ellas. Estos algoritmos optimizan una población de soluciones con el fin de obtener las soluciones de mejor calidad para el problema. Sin embargo, la tendencia natural de los algoritmos genéticos es siempre converger a la mejor solución, por lo que suelen presentar poca diversidad en el conjunto final de soluciones. Debido a esto, la búsqueda y el mantenimiento de múltiples soluciones en la población es un reto para el uso de algoritmos genéticos en problemas multi-modales, es decir, problemas con varios óptimos globales.

La extracción de reglas de asociación constituye un problema altamente multimodal en el que todas las reglas de mejor calidad (las soluciones óptimas del problema) deben ser obtenidas debido a que proporcionan interesantes y diferentes conocimientos de la base de datos. Los algoritmos genéticos basados en nichos [DMQS11] han demostrado ser un método interesante para tratar problemas altamente multi-modales, ya que estos algoritmos permiten localizar y mantener varios óptimos globales evitando la convergencia a una sola solución. Por esta razón, el uso de los algoritmos genéticos basados en nichos en el problema de extracción de reglas de asociación puede representar una interesante manera de tratarlo, donde la semilla de cada nicho obtenido representa un regla de asociación

interesante que ofrece un conocimiento diferente del resto de las reglas.

En esta memoria centramos nuestra atención en el diseño de algoritmos evolutivos que permitan obtener reglas de asociación con mayor calidad y diversidad.

B Objetivos

El objetivo general de esta memoria es proponer algoritmos evolutivos para la obtención de reglas de asociación cuantitativas positivas y negativas con mejores propiedades de calidad y diversidad en el conjunto de reglas obtenido. En una fase inicial se estudia el diseño de algoritmos multi-objetivos para la extracción de reglas de asociación que integre distintas medidas de calidad. En segundo lugar se aborda el problema de obtener un conjunto de reglas de asociación de alta diversidad apoyándonos en los algoritmos genéticos basados en nichos.

Los objetivos específicos en los que se desglosa el objetivo general serían:

- Diseñar una variante evolutiva multi-objetivo para la extracción de reglas de asociación cuantitativas positivas que integre distintas medidas de calidad de las reglas.

- Diseñar un enfoque evolutivo multi-objetivo para extraer un conjunto reducido de reglas de asociación cuantitativas positivas y negativas de buena calidad y con un alto cubrimiento de la base de datos.

- Mejorar la diversidad de las reglas obtenidas mediante el diseño de un algoritmo genético basado en nichos.

Además, se propone realizar un nuevo módulo para la herramienta KEEL [AFSG+09] con una gran cantidad de algoritmos para la extracción de reglas de asociación que han sido publicados en la literatura, donde también se encuentren los algoritmos presentados en esta memoria. Este módulo permitirá facilitar el acceso a estos algoritmos para que puedan ser usados por cualquier usuario.

C Resumen

Para desarrollar los objetivos propuestos, la memoria está organizada en cuatro capítulos, una sección de comentarios finales y dos apéndices. La estructura de cada una de estas partes se introduce brevemente a continuación.

En el Capítulo 1, introducimos algunas definiciones básicas de las reglas de asociación y de algunas de sus medidas de calidad. Además presentamos un estudio sobre los algoritmos evolutivos para la extracción de reglas de asociación cuantitativas y los algoritmos genéticos basados en nichos existentes en la literatura. Finalmente, se describen las principales características de la herramienta KEEL [AFSG+09] y su empleo para la extracción de reglas de asociación.

En el Capítulo 2, presentamos un algoritmo evolutivo multi-objetivo para extraer reglas de asociación cuantitativas positivas con un buen equilibrio entre interpretabilidad y precisión. Además se describe el estudio experimental realizado y los resultados obtenidos para evaluar el rendimiento de esta propuesta. Finalmente, se presenta un breve resumen del capítulo.

En el Capítulo 3, se propone un algoritmo evolutivo para obtener reglas de asociación cuantitativas positivas y negativas de alta calidad con un buen equilibrio entre el número de reglas y el cubrimiento de la base de datos. Además se muestran los resultados experimentales obtenidos sobre diferentes bases de datos del mundo real para analizar el funcionamiento de nuestra propuesta. Por último, se presenta un breve resumen del capítulo.

En el Capítulo 4, se propone un algoritmo genético basado en nichos para obtener un conjunto diverso de reglas de asociación fáciles de entender, interesantes y con un buen cubrimiento de la base de datos. Además, se presenta un estudio experimental sobre varias bases de datos reales para evaluar la efectividad de esta propuesta. Finalmente, se presenta un resumen del capítulo.

Se ha incluido una sección de "Comentarios Finales", que resume los resultados obtenidos en esta memoria, presentando algunas conclusiones sobre éstos y se comentan algunos aspectos sobre trabajos futuros.

Finalmente, se incluyen tres apéndices. Los dos primeros se dedican a introducir las características de los algoritmos genéticos y los algoritmos evolutivos multi-objetivos utilizados en los capítulos 2 y 3. El tercer apéndice muestra los resultados experimentales obtenidos para evaluar el rendimiento de los métodos presentados. La memoria termina con una recopilación bibliográfica que recoge las contribuciones más destacadas en la materia estudiada.

Capítulo 1

Reglas de Asociación y Algoritmos Evolutivos

Las reglas de asociación cuantitativas (RACs) son obtenidas a partir de bases de datos (BD) con datos numéricos. Este tipo de reglas resultan de gran utilidad debido a que los datos en las aplicaciones del mundo real normalmente están compuestos por valores numéricos. Esto ha provocado que el diseño de nuevos algoritmos que permitan tratar con diferentes tipos de datos sea un reto para los investigadores de este campo. Sin embargo, la mayoría de estos métodos solo pueden extraer reglas de asociación positivas sin tener en cuenta las reglas de asociación las negativas que que también son interesantes al expresar la ausencia un conjunto de ítems ante la presencia de otros. Además, este tipo de reglas permite representar conocimiento para el que sería necesario varias reglas positivas, reduciendo el número de reglas necesarias. Esto ha provocado que algunos métodos hayan sido propuestos para extraer reglas de asociación positivas y negativas.

Muchos de los algoritmos evolutivos propuestos para extraer reglas de asociación han utilizado las medidas clásicas de soporte y confianza para medir la calidad de las reglas. Sin embargo, estas medidas presentan varios problemas y no describen completamente la calidad del conocimiento descubierto. Por esto, estos algoritmos no describen completamente la calidad del conocimiento descubierto en los datos. La necesidad de incluir nuevas medidas de calidad implica la trans-

formación del problema hacia un problema multiobjetivo que permita optimizar varias medidas a la vez. Esto ha llevado a que la aplicación de los algoritmos evolutivos multi-objetivos (AEMOs) a la extracción de reglas de asociación sea un tema importante en la comunidad investigadora. Varios trabajos han sido publicados sobre este tema [MMBC14] y actualmente varios problemas siguen abiertos.

Por otra parte, la calidad de las reglas también puede verse afectada por la similitud que exista entre ellas. En este sentido el uso de los algoritmos genéticos basados en nichos (AGNs) [GG13, Mah95] permitirá promover la obtención de una población de reglas más diversa que puede aportar una mayor calidad y variedad al conocimiento obtenido.

El capítulo se organiza en las siguientes secciones. La sección 1.1 introduce los conceptos generales de minería de datos. La sección 1.2 presenta las definiciones generales de reglas de asociación, prestando especial atención a las RAC positivas y negativas (RACPNs) y a sus medidas de calidad. La sección 1.3 presenta un breve resumen de las características de los algoritmos clásicos que se han publicado para extraer dichas reglas. En la sección 1.4 se presenta un estudio del estado del arte de los algoritmos evolutivos para la extracción de RACs, incluyendo un breve estudio sobre los AGNs que han sido presentados en la literatura. Finalmente, en la sección se presentan las principales características de la herramienta KEEL [AFSG$^+$09] y su empleo para la extracción de reglas de asociación.

En este capítulo, no se ha incluido una descripción detallada de los algoritmos genéticos y los AEMOs, relegando un estudio más profundo a los Apéndices A y B respectivamente.

1.1. Minería de Datos

Los avances en las tecnologías para el almacenamiento de datos han provocado el aumento de grandes cantidades de datos en las organizaciones. Este crecimiento ha conllevado el desarrollo de nuevas técnicas, como la minería de datos, que permitan realizar diferentes análisis de la información para apoyar la toma de decisiones.

Sobre la definición de minería de datos se han presentado varios criterios. Algunos autores [TM05, WFH11] coinciden en que es “el proceso de extraer patrones

de los datos y convertirlos en conocimiento útil, previamente desconocido." Han and Kamber [HK06] la definen como "la extracción o "minería" de conocimiento de grandes cantidades de datos". En general la minería de datos pretende extraer conocimiento interesante a partir de conjuntos de datos grandes y complejos.

La minería de datos constituye una de las fases del "Proceso de Descubrimiento de Conocimientos en los Datos", el cual no solo incluye la obtención de modelos o conjunto de patrones, sino que también se refiere a la evaluación y posible interpretación de estos modelos [HK06]. En la Figura 1.1 se describen los pasos de este proceso.

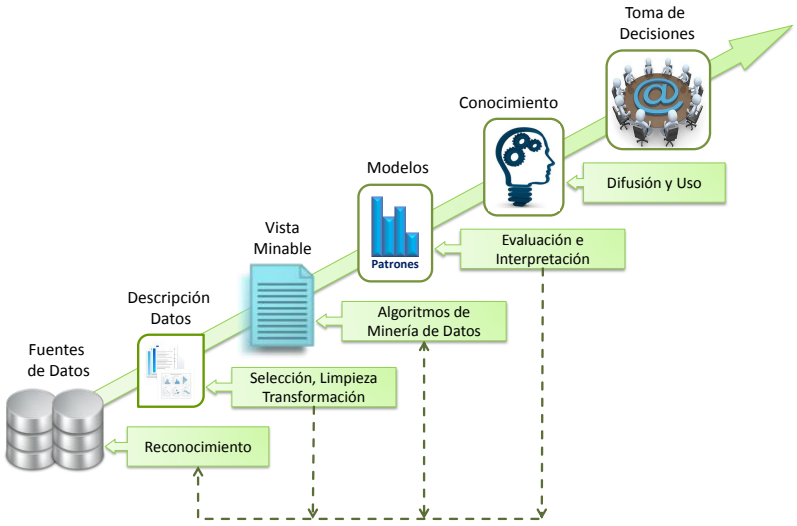

Figura 1.1: Pasos del proceso de descubrimiento de conocimiento en los datos

Las tareas de minería de datos permiten especificar el tipo patrón que puede ser extraído de los datos. De manera general, las tareas de minería de datos se pueden clasificar en supervisadas, semi-supervisadas o no supervisadas. Las tareas supervisadas [WFH11] son aplicadas sobre datos utilizando ejemplos etiquetados (el valor del objetivo para cada ejemplo es conocido), por otra parte las tareas semi-supervisadas [ZGBD09] consideran tanto ejemplos etiquetados como no etiquetados y las tares no supervisadas caracterizan las propiedades generales de los datos, donde los ejemplos no se encuentran etiquetados.

Los autores Han y Kamber [HK06] describen las tareas de minería de datos y los tipos de patrones que se pueden descubrir como sigue:

- *Descripción y caracterización de los datos*: representa un resumen de las características generales o características de una clase específica de los datos. Su aplicación resulta de gran utilidad para realizar un análisis descriptivo previo a la aplicación de las restantes actividades o tareas de minería de datos con el propósito de estudiar las características del conjunto de datos. Esta actividad puede efectuarse mediante variados métodos, entre los más simples para lograr un análisis descriptivo se encuentran los siguientes: ejecutar consultas SQL, OLAP (*On-Line Analytical Processing*). La salida de esta tarea se puede presentar en diversas formas, por ejemplo: gráficos circulares, gráficos de barras, curvas, cubos de datos multi-dimensionales, etc.

- Extracción de patrones frecuentes, asociaciones y correlaciones: representan patrones de comportamiento entre los datos en función de la aparición conjunta de valores de dos o más atributos que aparecen frecuentemente. La extracción de patrones frecuentes permite descubrir asociaciones y correlaciones interesantes dentro de los datos.

- Clasificación y predicción: la clasificación representa el proceso de encontrar un modelo (o función) que describe y distinga las clases de datos o conceptos, donde el modelo debe ser capaz de predecir la clase del objeto cuya etiqueta de clase es desconocida. El modelo se basa en el análisis de ejemplos de entrenamiento, es decir, ejemplos donde se conoce la etiqueta de su clase. Mientras que en la clasificación se predice una clase categórica (discreto, sin ordenar), en la predicción se predice un valor continuo.

- *Agrupamiento (Clustering)*: representa grupos de objetos de clases similares, donde no se necesita que los ejemplos estén etiquetados. Se obtienen grupos o conjuntos entre los elementos de tal manera que tengan una "similitud" alta entre ellos y baja con respecto a los elementos de otros clusters. Cada grupo que se forma puede ser visto como una clase de objetos, a partir del cual se pueden derivar reglas.

- *Análisis de valores atípicos*: representa el análisis de los objetos de datos que no cumplan con el comportamiento general, o el modelo de los datos, conocidos como objetos de datos atípicos. La mayoría de las técnicas de

minería de datos no tienen en cuenta este tipo de objetos, sin embargo en aplicaciones como la detección de fraudes, los eventos poco frecuentes pueden ser más interesantes que los que ocurren con más regularidad.

- *Análisis de la evolución de los datos*: describe y modela regularidades o tendencias para aquellos objetos cuyo comportamiento cambia en el tiempo. Los rasgos distintivos de este tipo de análisis incluyen el análisis de datos de series temporales, análisis de datos basados en similitud, entre otros.

El conocimiento que se obtiene al aplicar estas técnicas de minería de datos puede representarse de muchas maneras en dependencia de la técnica que se aplique. Esta representación del conocimiento constituye el modelo de los datos analizados. Los modelos se pueden representar mediante reglas, árboles de decisión, grupos o clusters, o sea en función de la tarea que cada uno de ellos representa.

La obtención de reglas de asociación es una técnica muy aplicada actualmente en diversos ámbitos. A continuación se describen los elementos básicos relacionados con esta técnica de minería de datos.

1.2. Reglas de Asociación

La obtención de reglas de asociación consiste en encontrar un modelo que describa dependencias significantes (o asociaciones) entre ítems de una BD [HK06]. Una de las aplicaciones más conocidas es el análisis de listas de mercado, porque puede ser semejante al análisis de artículos que frecuentemente se reúnen en la lista de compradores en un mercado [AIS93].

Las reglas son expresiones del tipo $X \rightarrow Y$, donde X y Y son conjuntos de ítems (parejas atributo-valor) y cumplen que $X \cap Y = \oslash$. Esto significa que si todos los ítems de X están en un ejemplo de la BD, entonces todos los ítems de Y están también en el ejemplo con una alta probabilidad, donde X y Y no tienen ningún ítem en común [AIS93].

Las medidas soporte y confianza son las más utilizadas para evaluar las RACs. Estas medidas se basan en el soporte de un conjunto de ítems I, el cual se define como:

$$SOP(I) = \frac{\mid \{e \in D \mid I \in e\} \mid}{\mid D \mid} \tag{1.1}$$

donde el numerador es el número de ejemplos de la BD cubiertos por I, y $\mid D \mid$ es el número de ejemplos de la BD.

Luego, el soporte y la confianza de una regla $X \rightarrow Y$ se define como:

$$soporte(X \rightarrow Y) = SOP(XY) \tag{1.2}$$

$$confianza(X \rightarrow Y) = \frac{SOP(XY)}{SOP(X)} \tag{1.3}$$

donde SOP(XY) es el soporte de la reglas completa y SOP(X) es el soporte del antecedente de la regla.

Las técnicas clásicas para extraer reglas de asociación intentan obtener reglas con valores de soporte y confianza mayores que un mínimo de soporte y una mínima confianza. Sin embargo, varios autores han señalado algunos inconvenientes de este marco de trabajo que conduce a encontrar reglas de baja calidad [BBSV02]. La confianza no detecta independencia estadística o dependencia negativa entre los ítems debido a que no tiene en cuenta el soporte del consecuente. Por ejemplo, una confianza de 0,9 en una regla $A \rightarrow B$ no es mejor que una confianza de 0,7 en la regla $C \rightarrow D$ si el soporte de B es 0,95 y el soporte de D es 0,1, ya que la primera de las reglas se corresponde con una dependencia negativa (observando A se reduce la probabilidad de B), mientras que en la segunda la probabilidad de D incrementa significativamente cuando observamos C. Asimismo, los ítems con soporte muy alto pueden generar reglas de baja calidad porque cualquier conjunto de ítems parece ser un buen predictor de ellos. Por esta razón, en la literatura se han propuesto otras medidas de calidad para la selección y ranking de ejemplos de acuerdo con su interés potencial para el usuario [GH06]. En la próxima subsección describimos varias de las características de las medidas de interés del estado del arte que han sido propuestas para medir la calidad de las reglas.

1.2.1. Medidas de interés

Con el fin de mejorar la calidad de las reglas que son producidas por los algoritmos, los usuarios podrían usar fórmulas más complejas para determinar si una regla es interesante o no. El interés de una regla se ha definido como un concepto amplio que abarca varias características: concisa, general, fiable, peculiar, novedosa, sorprendente, diversa, útil y accionable [GH06]. A partir de

esta definición, Geng y Hamilton presentaron dos categorías para clasificar las medidas de interés: objetivas y subjetivas [GH07] .

Las medidas *objetivas* solo se basan en los ejemplos de la BD, por lo que no requieren ningún conocimiento adicional sobre los datos. Los factores de concisión, generalidad, fiabilidad, peculiaridad y diversidad dependen sólo de los datos y los patrones, por lo tanto pueden ser considerados como objetivos. La mayoría de las medidas objetivas se basan en las teorías de la probabilidad, estadística, o teoría de la información. Por el contrario, las medidas *subjetivas* tienen en cuenta los datos y el usuario que usa estos datos. Una medida de este tipo requiere el conocimiento del usuario acerca de los datos. En este sentido se consideran subjetivas las características de novedosa, sorprendente, útil y accionable porque dependen no solo de los datos y los patrones, sino también de la persona que utilice las reglas obtenidas.

En esta memoria nos concentramos en el estudio de las medidas de interés objetivas porque no necesitan el conocimiento de un usuario sobre los datos. A continuación describimos brevemente las medidas basadas en probabilidad que hemos utilizado en esta memoria, las cuales evalúan la generalidad y fiabilidad de las reglas obtenidas, aunque muchas más pueden ser encontradas en [GH06, BBSV02, Gla13].

- La medida *conviction* [BMUT97] mide la dependencia entre X y $\neg Y$, donde $\neg Y$ significa ausencia de Y. Esta medida toma valores en el intervalo [0,∞), donde valores menores que 1 representan dependencia negativa, 1 representa independencia y mayores que 1 representan dependencia positiva. Sin embargo, no es fácil utilizar los valores de esta medida para comparar las reglas, ya que su dominio no está limitado, por lo que es difícil definir un umbral para ella. Conviction de una regla $X \rightarrow Y$ se define como:

$$conviction(X \rightarrow Y) = \frac{SOP(X)SOP(\neg Y)}{SOP(X\neg Y)} \tag{1.4}$$

- La medida *lift* [RMS98] representa el ratio entre la confianza de la regla y la confianza esperada de la regla. Al igual que conviction, esta medida toma valores en el intervalo [0,∞) donde valores menores que 1 representan dependencia negativa, 1 representa independencia y mayores que 1 representan dependencia positiva. Lift de una regla $X \rightarrow Y$ se define como:

$$lift(X \rightarrow Y) = \frac{Confidence(X \rightarrow Y)}{SOP(Y)} = \frac{SOP(XY)}{SOP(X)SOP(Y)} \tag{1.5}$$

- El *factor de certeza* (FC) [SB75] mide la variación de la probabilidad de que Y esté en un ejemplo cuando se consideran solo los ejemplos donde está X. FC toma valores en el intervalo [-1,1] donde valores positivos y negativos representan dependencia positiva y negativa respectivamente y 0 representa independencia. Esta medida para una regla $X \rightarrow Y$ se define de tres maneras dependiendo de si la confianza es menor, mayor o igual que $SOP(Y)$:

  Si $confianza(X \rightarrow Y) > SOP(Y)$

  $$FC(X \rightarrow Y) = \frac{confianza(X \rightarrow Y) - SOP(Y)}{1 - SOP(Y)} \tag{1.6}$$

  Si $confianza(X \rightarrow Y) < SOP(Y)$

  $$FC(X \rightarrow Y) = \frac{confianza(X \rightarrow Y) - SOP(Y)}{SOP(Y)} \tag{1.7}$$

  Sino es 0

- La medida *Netconf* [AK04] evalúa una regla basándose en el soporte de la regla, del antecedente y del consecuente. Netconf obtiene valores en el intervalo [-1,1] donde valores positivos y negativos representan dependencia positiva y negativa respectivamente y 0 representa independencia. Netconf de una regla $X \rightarrow Y$ se define como:

  $$netconf(X \rightarrow Y) = \frac{SOP(XY) - SOP(X)SOP(Y)}{SOP(X)(1 - SOP(X))} \tag{1.8}$$

- La medida *yule'sQ* [TKS02] representa la correlación entre dos eventos posiblemente relacionados. Esta medida obtiene valores en el intervalo [-1,1], donde 1 implica una correlación positiva perfecta, -1 implica una correlación negativa perfecta y 0 implica que no hay correlación. Esta medida satisface casi todas las propiedades de las medidas de interés que han sido propuestas en la literatura [GH06, PS91]. Yule'sQ para una regla $X \rightarrow Y$ se define como:

  $$\frac{SOP(XY)SOP(\neg X \neg Y) - SOP(X \neg Y)SOP(\neg XY)}{SOP(XY)SOP(\neg X \neg Y) + SOP(X \neg Y)SOP(\neg XY)} \tag{1.9}$$

Por otra parte, también se han presentado algunos estudios para evaluar la consición y peculiaridad de las reglas obtenidas, basados en la representación de las reglas, es decir, en la información que ellas expresan a los usuarios [GH07]. Los

estudios relacionados con la consición [GH07] se refieren a la cantidad de ítmes que involucra una regla o al tamaño de un conjunto de reglas, donde analizar menos ítems en una regla y menos número de reglas en un conjunto resulta más comprensible e interesante para los usuarios. La mayoría de los trabajos se han concentrado en evitar redundancia en el conjunto de reglas obtenido [BPT$^+$00, PT00, GH07].

Las medidas para evaluar la peculiaridad [GH07] de una regla se basan en su lejanía al resto de las reglas, utilizando una función de distancia para medir dicha lejanía. En este sentido se han presentado varios estudios [GH07], un ejemplo de ellos es la medida *neighborhood-based unexpectedness* [DL98], la cual define el grado de interés de una regla basado en la imprevisibilidad de su vecindad. Para ello define una función de distancia entre las reglas a partir de calcular la diferencia entre los ítems involucrados en las reglas, teniendo en cuanta por separado la diferencia entre todos los ítems involucrados, entre los del antecedente y entre los del consecuente de dos reglas. Además define la vecindad de una regla, donde dos reglas serán vecinas si su distancia es menor que un umbral. A partir de estos conceptos propone dos medidas para evaluar el interés de una regla: confianza inesperada y otra medida basada en la dispersión de la vecindad de una regla.

Destacar que las medidas de interés han tenido importantes usos en el proceso de extracción de reglas de asociación [GH06]. En primer lugar, han sido utilizadas para identificar patrones poco interesantes durante el proceso de minería y así reducir el espacio de búsqueda, mejorando la eficiencia de este proceso. También se han utilizado para realizar ranking entre las reglas y en fases de post-procesamiento para proveer solo reglas interesantes.

Muchas de las propiedades de este tipo de medidas se han examinado y comparado en la literatura, ya sea para ayudar a la selección de una medida apropiada para un contexto determinado o para determinar sus méritos teóricos o computacionales [GH06, Gla13, PS91]. Recientemente, Glass presentó en [Gla13] un nuevo estudio sobre las propiedades que deben cumplir estas medidas, identificando tres nuevas propiedades claves: simetría, maximalidad/minimalidad y monotonía. Este trabajo muestra que ninguna de las medidas presentadas en la literatura satisfacen todas las propiedades por lo que propone dos nuevas medidas (ks y $k\phi$). Destacar que también considera a FC [SB75] como una de las mejores medidas presentadas al cumplir con la mayoría de las propiedades.

Finalmente, varios autores [GH07, BJQ$^+$14] han presentado diferentes méto-

dos de selección y agregación de las medidas de interés. Los métodos de selección permiten a los usuarios filtrar las medidas de interés redundantes para utilizar un conjunto más pequeño de medidas y los métodos de agregación pueden ser utilizados para integrar las preferencias de los usuarios utilizando pesos para combinar los valores de un conjunto más pequeño de medidas. En [BJQ+14] se presenta el proceso general para la selección apropiada de una medida de interés:

- Definición de las propiedades requeridas y su peso (importancia). Las propiedades podrían ser las propiedades del conjunto de datos, propiedades propias de las medidas de interés o incluso los valores producidos por las medidas en un conjunto de datos de prueba.
- Cuantificar la similitud y diferencias entre las propiedades y las medidas. Las medidas que tienen las mismas propiedades se consideran redundantes (utilizando una es aproximadamente lo mismo que usarlas todas) y se pueden eliminar.
- Analizar el equilibrio de las medidas con dichas propiedades. Las medidas que mejor se ajusten serán seleccionadas.

1.2.2. Reglas de Asociación Cuantitativas Positivas y Negativas

Muchos trabajos previos para la extracción de reglas de asociación han sido enfocados sobre BDs con valores binarios, sin embargo las BDs en las aplicaciones del mundo real están compuestos normalmente por valores cuantitativos. Debido a ello, varios estudios han sido presentados para extraer RACs a partir de BDs con valores cuantitativos, donde cada ítem es un par *atributo-intervalo* [SA96]. Por ejemplo, una RAC positiva es $Salario \in [3000, 3500] \rightarrow NumCoches \in [3, 4]$.

Muchos de estos métodos solo consideran ítems positivos en la extracción de reglas de asociación sin tener en cuenta los negativos, los cuales también son interesantes al expresar por ejemplo la presencia de X ante la ausencia de Y ($X \rightarrow \neg Y$) [SBM98], ofreciendo un nuevo conocimiento para apoyar la toma de decisiones. Las reglas negativas al igual que las reglas positivas consideran un conjunto de ítems, pero además, pueden incluir ítems negativos dentro del antecedente ($\neg X \rightarrow Y$), del consecuente ($X \rightarrow \neg Y$), o de ambos ($\neg X \rightarrow \neg Y$). Por otra parte las reglas positivas solo incluyen ítems positivos, mientras que las reglas de asociación negativas deben incluir al menos un ítem negativo. La Figura

1.2 muestra el dominio del ítem positivo $Altura \in [90, 150]$ y del ítem negativo $Edad \in \neg[5, 25]$.

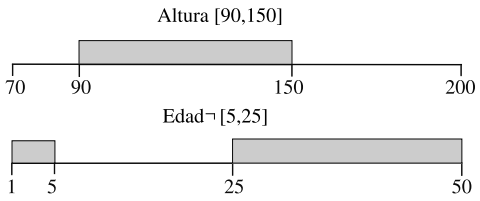

Figura 1.2: Ejemplo de un ítem positivo y un ítem negativo

Destacar que, el uso de ítems negativos en la extracción de RACs permite reducir el número de reglas necesarias para extraer conocimiento interesante de las BDs. Esto es debido a que un ítem negativo permite representar conocimiento para el que se necesitarían varios ítems positivos para representarlo. La Figura 1.3 muestra un ejemplo de como se necesita utilizar los ítems positivos $X \in [1, 3]$ y $X \in [6, 10]$ para expresar la misma información que puede representar el ítem negativo $X \in \neg[3, 6]$. Esto ha provocado que durante los últimos años algunos métodos hayan sido propuestos para extraer reglas de asociación positivas y negativas [SBM98, AA06, TRD12].

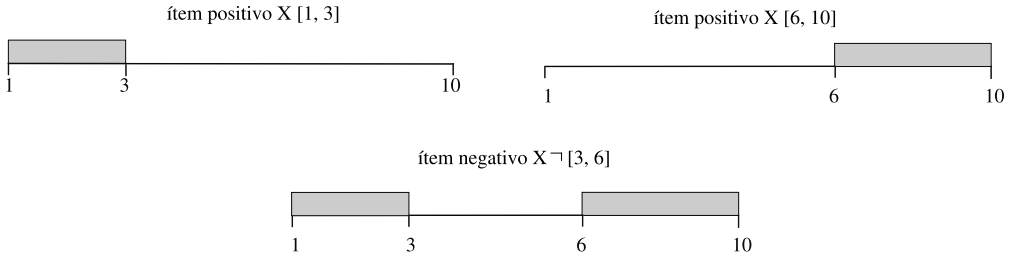

Figura 1.3: Ejemplo del conocimiento representado por dos ítems positivos y un ítem negativo

1.3. Algoritmos clásicos de extracción de reglas de asociación

Inicialmente, los métodos de extracción de reglas de asociación se centraron en la identificación de asociaciones dentro de BD de valores categóricos. La mayoría de estos métodos seguían un esquema basado en dos etapas, donde en la primera etapa se identifican todos los conjuntos de items frecuentes que existían en la BD, y en la segunda etapa se extraen todas las reglas de asociación cuya confianza supere un valor mínimo de confianza proporcionado por el experto. Destacar que un conjunto de items es considerado frecuente si su soporte supera un valor mínimo de soporte proporcionado por el experto [AIS93].

Algunos de los métodos clásicos más conocidos para la extracción de reglas de asociación son Apriori [SA96], Eclat [Zak00] y FP-growth [HPYM04]. A continuación describiremos las características principales de cada uno de ellos:

Apriori [SA96] es considerado el algoritmo de aprendizaje básico de reglas de asociación. En cada iteración genera conjuntos de ítems candidatos a partir de los conjuntos de ítems que han sido considerados frecuentes en la iteración anterior. Una vez extraídos todos los conjuntos de ítems frecuentes, Apriori genera a partir de ellos todas las reglas con una confianza mayor a un umbral definido por el usuario. Este algoritmo presenta dos limitaciones importantes: genera un gran número de ítems candidatos y necesita escanear la BD repetidas veces para calcular el soporte de los ítems candidatos [HPYM04]. Estos problemas provocan que este método necesite una gran cantidad de espacio en memoria y de tiempo computacional cuando el tamaño de la BD aumenta un poco.

Por otra parte, Eclat [Zak00] emplea una estrategia de primero en profundidad. Este método primero genera los ítems candidatos, extendiendo los prefijos de un conjunto de ítems hasta que se encuentre uno que no sea frecuente. En tales casos, simplemente se da marcha atrás al prefijo anterior y luego se aplica de forma recursiva el procedimiento anterior. A diferencia de Apriori, para todos los ítems de una BD, primero se construye una lista con todos los identificadores de las transacciones que contienen ese ítem. Luego, se calcula el soporte cruzando dos o más listas de identificadores para comprobar si tienen ítems en común. Si es así, el soporte es igual al tamaño del conjunto resultante. El proceso para la generación de las reglas positivas es el mismo que Apriori.

Han y colaboradores propusieron FP-growth [HPYM04] para solucionar las limitaciones de Apriori y Eclat, presentando un algoritmo que permite extraer

ítems frecuentes sin necesidad de generar conjuntos de ítems candidatos. Primero, este algoritmo construye una estructura de datos llamada árbol "patrón-frecuente"(FP-tree) para almacenar información cuantitativa sobre los patrones frecuentes. Para asegurarse de que la estructura del árbol sea compacta e informativa, los nodos del árbol solo representarán los ítems frecuentes de tamaño 1. De esta manera, para extraer los siguientes ítems frecuentes solo se necesitará trabajar con el FP-tree, en vez de con toda la BD. Además, la técnica de búsqueda empleada para extraer los patrones frecuentes se basa en el particionamiento (*partitioning-based*), basada en la idea de divide y vencerás, en lugar de generar todas las combinaciones posibles de los ítems frecuentes (generación de candidatos) como hace Apriori. Sin embargo, esta propuesta también presenta problemas, aumentando su costo computacional cuando aumenta el tamaño de la BD. En la literatura se han presentado diferentes extensiones para mejorar este método [dJBC10].

Un reto importante en la extracción de ítems frecuentes en grandes BDs es la generación de un gran número de ítems que satisfacen el mínimo de soporte, sobre todo cuando el valor mínimo que se establece es muy bajo. La tarea de buscar conjuntos de ítems frecuentes es costosa, ya que los ítems al ser analizados crecen exponencialmente con respecto al número de variables y valores distintos que presenten los datos. En esta situación, generar todos los ítems frecuentes no es muy buena idea. Para solucionar este problema, se han presentado en el estado del arte diferentes soluciones [Goe02] para extraer solo un subconjunto de items frecuentes, dos de las propuestas más conocidas son la extracción de patrones frecuentes "cerrados"(*closed frequent pattern*) [Bay98] y patrones frecuentes "maximales"(*maximal frequent pattern*) [PBTL99]. Un conjunto de ítems frecuentes cerrado representa un conjunto donde no existen super conjuntos inmediatos que tengan el mismo soporte que él, mientras que los ítems frecuentes maximales representan un conjunto de ítems que no tienen super conjuntos frecuentes (sin importar si tienen menor o mayor soporte). Por esto, un conjunto de ítems frecuentes cerrados contiene la información completa respecto a sus ítems frecuentes correspondientes (mismo soporte) y el conjunto de ítems maximales, aunque representa una información más compacta, por lo general no contiene la información respecto a sus correspondientes ítems frecuentes.

El problema de la extracción de ítems frecuentes aumenta en la extracción de RACs, debido a que los atributos cuantitativos contienen muchos valores distintos. Para evitar dicho problema un método normalmente utilizado es dividir el dominio de los atributos numéricos en intervalos, lo cual es un problema críti-

co en la extracción de RACs porque la información no está clasificada y dichos intervalos pueden tener una gran influencia sobre las reglas obtenidas.

1.4. Algoritmos Evolutivos

En los últimos años, varios investigadores han propuesto algoritmos evolutivos [ES03] para aprender los intervalos de las RACs [AFFPBH10]. La principal motivación de utilizar los algoritmos evolutivos en este problema es porque son considerados una de las técnicas de búsqueda más exitosas para problemas complejos y han demostrado ser muy buenos en el aprendizaje y la extracción de conocimiento.

Los métodos de aprendizaje evolutivo siguen dos enfoques distintos para codificar reglas en una población de individuos:

- "Cromosoma = Conjunto de reglas", o enfoque Pittsbourgh, en el que cada individuo representa un conjunto de reglas [Smi80]. En este caso, los cromosomas evolucionan bases de reglas completas, y compiten entre sí durante la búsqueda.
- "Cromosoma = Regla", en el que cada individuo representa una única regla, y el conjunto de reglas completo se obtiene combinando varios individuos de una población (cooperación) o a partir de varias ejecuciones (competición).

  Dentro el enfoque "Cromosoma = Regla", existen:

  - El enfoque Michigan, en el que cada individuo codifica una única regla. Este tipo de sistemas suelen denominarse sistemas de aprendizaje de clasificadores [HR77]. Son sistemas de paso de mensajes, basados en reglas que emplean aprendizaje por refuerzo y un AG para aprender reglas que guíen el aprendizaje en un problema dado. El AG detecta nuevas reglas que reemplazan a las peores mediante competición entre cromosomas en el proceso evolutivo.
  - El enfoque IRL (*Iterative Rule Learning*, aprendizaje iterativo de reglas), en el que cada cromosoma representa una regla. Los cromosomas compiten entre sí, tomándose el mejor en cada ejecución del AG. La solución global se forma a partir de ellos, ejecutando múltiples veces el algoritmo.
  - El enfoque GCCL (*Genetic Cooperative-Competitive Learning*, aprendizaje genético cooperativo-competitivo), en el que la base de reglas se

codifica a partir de la población completa o de un subconjunto de ella. En este modelo, los cromosomas compiten y cooperan simultáneamente.

Todos ellos han sido empleados para el aprendizaje de base de reglas. La selección de uno de los dos enfoques depende del tipo de regla que se quiera obtener [GJ05].

En la literatura se han presentado diferentes tipos de algoritmos evolutivos para resolver un gran número de problemas, un ejemplo de ellos son: los algoritmos genéticos [Gol89], los AEMOs [CLV02] y los AGNs [Mah95].

Los algoritmos genéticos representan un tipo de algoritmo evolutivo de búsqueda general que se basa en principios inspirados en la genética de las poblaciones naturales para llevar a cabo un proceso evolutivo sobre soluciones de problemas. Estos algoritmos han tenido mucho éxito en problemas de búsqueda y optimización porque tienen la habilidad para explotar la información que van acumulando sobre el espacio de búsqueda que manejan, desconocido inicialmente, lo que les permite redirigir posteriormente la búsqueda hacia subespacios útiles.

Los AEMOs se conocen como aquellos algoritmos genéticos en los que se definen múltiples objetivos relevantes para un mismo problema que, en general, están en conflicto. Por lo que se definen múltiples funciones de adaptación a evaluar, cada una de ellas asociada a un objetivo diferente.

Por otra parte, los AGNs extienden a los algoritmos genéticos para localizar y mantener múltiples soluciones óptimas en la población para resolver problemas con varios óptimos globales.

En las siguientes subsecciones de esta memoria veremos algunos de los algoritmos genéticos y los AEMOs más representatativos para la extracción de RACs. Además se presenta un estudio de los AGNs como vía de promover diversidad en las reglas obtenidas.

1.4.1. Algoritmos Géneticos

Uno de los primeros algoritmos genéticos para extraer RACs fue GENAR (*GENetic Association Rules*) [MAR01], presentado por Mata y colaboradores en el año 2001. En este algoritmo un cromosoma codifica una regla de asociación, que contiene los intervalos máximos y mínimos de cada atributo numérico. Sin embargo, cada regla involucra el total de atributos de la BD y solo el último atributo formará parte del consecuente, reduciendo considerablemente el conjunto

de reglas interesantes que puede encontrar el método. La función objetivo solo considera el número de ejemplos incluidos en la regla y penaliza las reglas que cubran ejemplos de la BD que también han sido cubiertos por otras reglas.

Otro método de los mismo autores de GENAR es GAR *Genetic Association Rules* [MAR02], un método que extiende GENAR [MAR01] para encontrar conjuntos de ítems frecuentes en BDs numéricas sin tener que discretizar los valores de los atributos. Cada cromosoma representa un conjunto de ítems, donde cada gen representa el máximo y el mínimo de los valores de los atributos que pertenecen al conjunto de ítems. Cuando el algoritmo termina el proceso evolutivo, se ejecuta otro procedimiento que debe ser generado para generar las reglas a partir de los conjuntos de ítems frecuentes extraídos. Recientemente, uno de los autores de GAR, presentó un algoritmo similar llamado GAR-plus [PAMV12], un algoritmo evolutivo que extraer directamente reglas de asociación, cuyo objetivo es encontrar reglas de asociación interesantes que se pueden extraer de una BD con atributos tanto numéricos como discretos.

QuantMiner [SAVN07] representa otro ejemplo del uso de los algoritmos genéticos para aprender los intervalos de las RACs, optimizando el soporte y la confianza de las reglas, mediante el uso de una función objetivo basada en la medida de interés *gain* [FMMT96]. Similar a los métodos anteriores, en este método un cromosoma también representa una regla de asociación, codificando para cada variable numérica los valores mínimo y máximo de su intervalo.

EARMGA (*Evolutionary Association Rules Mining with Genetic Algorithm*) [YZZ09] presentado por Yan en el año 2009 utiliza un algoritmo genético para identificar RACs, sin que el usuario especifique un valor mínimo de soporte. Cada cromosoma codifica una regla con una longitud especificada por el parámetro k. Las reglas más interesantes son obtenidas conforme al grado de interés definido por la función objetivo, el cual se basa en el soporte de la regla, el soporte del antecedente y el soporte consecuente.

Por otra parte, Alatas y colaboradores [AA06] diseñaron un algoritmo genético (lo llamaremos Alatasetal) para simultáneamente buscar los intervalos de los atributos cuantitativos y descubrir las reglas de asociación positivas y negativas asociadas a esos intervalos. Los cromosomas representan reglas de asociación, en los cuales cada gen tiene 4 partes. La primera parte representa si forma parte del antecedente o del consecuente de la regla, la segunda si el intervalo es positivo o negativo, la tercera y cuarta representan el límite inferior y superior del intervalo del atributo respectivamente.

QARGA (*Quantitative Association Rules by Genetic Algorithm*) [MBMATR11] fue diseñado para extraer reglas de asociación en series temporales. Este algoritmo genético busca los intervalos más adecuados para encontrar RACs con altos valores de soporte y confianza, junto con otras medidas utilizadas para medir la calidad de las reglas como: la amplitud de los intervalos y la cantidad de atributos involucrados en la regla. Cada individuo es representado por un arreglo de tamaño igual a la cantidad de atributos de la BD, donde cada gen codifica el límite mínimo y máximo del intervalo de un atributo y el tipo de atributo, el cual indica si el gen es considerado en la regla. Este algoritmo se basa en el proceso de aprendizaje iterativo de una regla, por lo que el algoritmo genético se aplica en cada iteración obteniendo una regla por iteración (el mejor individuo encontrado). Recientemente, este algoritmo fue utilizado para validar el diseño de una nueva función objetivo [MBMATR14] compuesta por la selección de un subconjunto de medidas de interés, las cuales fueron obtenidas a partir del análisis de componentes principales sobre un conjunto de medidas que evalúan la calidad de las RACs.

De manera general, la mayoría de los algoritmos presentados utilizan un enfoque Michigan para representar sus individuos, donde incluyen los valores mínimos y máximos de los intervalos de los atributos numéricos. Además, casi todos optimizan las medidas de soporte y confianza de las reglas, sin tener en cuenta las medidas de interés para evaluar la calidad de las reglas (descritas en la sección 1.2.1).

Finalmente, en [dJGGP11] se presenta un estudio de los algoritmos evolutivos presentados en la literatura para la extracción de reglas de asociación. Este estudio plantea dos aspectos importantes a tener en cuenta en el uso de los algoritmos genéticos para este tipo de problema:

- Debe utilizarse una combinación de diferentes medidas de calidad para evaluar las reglas, porque si sólo se consideran el soporte y/o confianza los algoritmos probablemente pueden caer en óptimos locales.
- Deben incorporarse mecanismos para preservar la diversidad porque de lo contrario el algoritmo convergerá a unas pocas reglas de asociación de alta calidad.

1.4.2. Algoritmos Evolutivos Multi-Objetivos

El proceso de extraer reglas de asociación puede ser tratado como un problema de optimización multi-objetivo más que como un problema mono-objetivo,

en el que las diferentes medidas utilizadas para evaluar las reglas puedan ser consideradas como distintos objetivos de este problema.

La mejor forma de resolver este tipo de problemas es mediante el uso de los criterios de dominancia y pareto-optimalidad. Una solución domina a otra si es mejor o igual en todos los objetivos y al menos mejor en uno de ellos. De esta forma, todas las soluciones que no son dominadas por ninguna otra solución se llaman pareto-optimal, existe u forma la frontera extracción de regl diferentes frentes

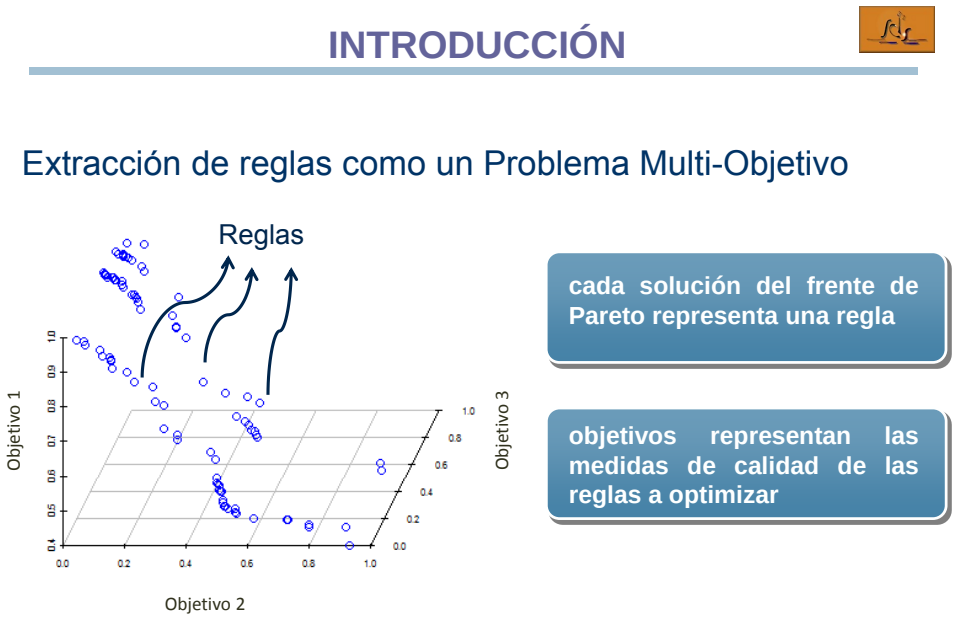

Figura 1.4: Ejemplo de frentes de Pareto en el problema de extracción de reglas de asociación

En los últimos años han habido avances significativos en el desarrollo de algoritmos evolutivos para problemas de optimización multi-objetivos. Los AEMOs tratan simultáneamente con un conjunto de posibles soluciones (llamado población), permitiéndoles encontrar varios miembros del conjunto optimal de Pareto en una sola ejecución del algoritmo. Además, estos algoritmos no son demasiado susceptibles por la forma o la continuidad del frente de Pareto (por ejemplo, pueden tratar fácilmente con frentes de Pareto discontinuos y cóncavos) [SR11].

En la última década, se han propuesto varios AEMOs para extraer reglas de asociación, donde cada solución del frente de Pareto representa una RAC.

Este enfoque elimina algunas de las limitaciones de los algoritmos mono-objetivo presentados en la sección anterior y permite optimizar varias medidas con el fin de extraer reglas de alta calidad.

Uno de los primeros AEMOs para extraer reglas de asociación fue propuesto por Ghosh y Nath en el año 2004 [GN04] (en esta memoria lo llamaremos MOEA_Ghosh). Este algoritmo utiliza un algoritmo genético basado en el frente de Pareto para extraer reglas de asociación útiles e interesantes para cualquier BD. Cada cromosoma representa una regla de asociación. Este método optimiza tres medidas: comprensibilidad, interés y presición. Además, emplea una población externa para almacenar las soluciones no dominadas encontradas.

Otro enfoque relevante en la optimización multi-objetivo es el algoritmo MODENAR (*Multi-objective differential evolution algorithm for mining numeric association rules*) [AA08] propuesto en el 2008. Este método utiliza un AEMO diferencial para extraer RACs precisas y comprensibles sin que el usuario especifique un mínimo de soporte y un mínimo de confianza. MODENAR utiliza una codificación similar a la presentada en [AA06], pero sin incluir la parte del gen donde se representa si el intervalo es positivo o negativo. Además, este método optimiza cuatro objetivos para mejorar la calidad de las reglas: soporte, confianza, comprensibilidad y amplitud del dominio de los intervalos.

ARMMGA (*Multi-objective association rules with genetic algorithms*) [QNMB11] es un AEMO basado en el método EARMGA [YZZ09] para extraer RACs, sin que el usuario especifique un mínimo de soporte y un mínimo de confianza. Según los comentarios de los autores, el aspecto más importante de este algoritmo es que su función objetivo solo especifica el orden de los cromosomas en la población y no tiene ningún otro efecto en los operadores del AG, utilizando este orden como criterio de selección. Este algoritmo devuelve la población con mejor promedio en los valores de la función objetivo, la cual se basa en el producto del soporte y la confianza de la regla.

Recientemente, Coello y colaboradores [MMBC14] presentaron un estudio sobre los diferentes AEMOs que han sido propuestos para extraer reglas de asociación. En este estudio se muestra que muy pocos AEMOs han sido diseñados para extraer RACs. Por otra parte, se presenta un resumen de las características de estos AEMOs, donde se evidencia que la mayoría codifican las reglas utilizando un enfoque Michigan, optimizan diferentes medidas de calidad, siendo el soporte y la confianza las medidas más utilizadas y obtienen como resultado final el conjunto de soluciones no dominadas encontradas.

1.4.3. Algoritmos Genéticos basados en nichos

Como se ha dicho, los algoritmos evolutivos en general, y en particular los algoritmos genéticos, han sido ampliamente utilizados para generar RACs. No obstante, como veremos en esta sección el uso de los algoritmos evolutivos puede afectar la diversidad del conjunto de reglas obtenido.

Los algoritmos evolutivos [ES03] tratan simultáneamente con un conjunto de posibles soluciones que les permite encontrar varias soluciones óptimas en una sola ejecución del algoritmo. Sin embargo, la tendencia natural de los algoritmos genéticos es siempre converger a la mejor solución y por lo general presentan poca diversidad en el conjunto final de soluciones. Por lo tanto, la búsqueda y el mantenimiento de múltiples soluciones en la población es un reto para el uso de los algoritmos genéticos en problemas multi-modales, es decir, problemas con varios óptimos globales. Debido a esto, han surgido los AGNs [PPH12, DMQS11], los cuales extienden a los algoritmos genéticos para localizar y mantener múltiples soluciones óptimas en la población para este tipo de problemas.

Uno de los primeros estudios para preservar diversidad fue la disertación de Cavicchio en 1970 [Cav70]. En este estudio se propusieron diferentes esquemas de preselección para la sustitución de un hijo por uno de sus padres, los cuales se basaron en la afirmación de que un hijo es generalmente similar a sus padres. Cinco años más tarde, De Jong generalizó estos esquemas de preselección con un nuevo método llamado *crowding* [DJ75]. La utilización de los esquemas de preselección o del método *crowding* en las funciones multi-modales no siempre genera buenos resultados, ya que pueden conservar varios representantes del mismo óptimo debido a los errores de sustitución que presentan [Mah95]. Para resolver este problema se han desarrollado muchas versiones del método *crowding*, siendo las más representativas las siguientes: *Crowding* Determinístico [Mah92], Selección por Torneo Restringida [Har94], *Crowding* Multi-Nicho [CV97] y *Crowding* Probabilístico [MG99].

En 1987, Goldberg y Richardson propusieron el método *sharing*, en el cual se disminuye la evaluación de los individuos de acuerdo con el número de individuos similares que tengan de la población [GR87]. Este método permite a los algoritmos genéticos trabajar simultáneamente con varios óptimos globales en problemas de optimización multi-modales. Sin embargo, estudios posteriores han mostrado algunas limitaciones debido al parámetro *sharing* que ellos utilizan y a su alta complejidad computacional [SK98]. En los años siguientes, varios algoritmos fueron diseñados con el fin de resolver ambos inconvenientes. La mayoría

de ellos se centraron en realizar una distribución preliminar de los individuos en nichos utilizando un algoritmo de agrupamiento o en penalizar la evaluación de los individuos a partir de su similitud sobre una muestra de individuos de la población de tamaño fijo [YG93, LW02, SK98, CKOS91].

Posteriormente, Petrowski propuso en [Pét96] el método *clearing* basado en el mismo concepto que el método *sharing*. Sin embargo, a diferencia de método *sharing*, en esta propuesta sólo sobreviven los mejores individuos de cada nicho, igualando a 0 la evaluación del resto de los individuos. Este proceso se aplica después del proceso de evaluación y antes de la selección. Siguiendo esta idea, se han presentado otros métodos que también modifican la calidad de las soluciones de acuerdo a su similitud con otros individuos [LCJ99, KCJL02].

Recientemente, Li presentó un enfoque diferente basado en nichos, llamado conservación de *especies* [LBPC02]. Esta propuesta divide la población en varias especies en función de su similitud y cada una de estas especies se conforma en torno a un individuo dominante, comúnmente conocido como la semilla de la especie. Esta técnica ha demostrado ser eficaz para obtener múltiples soluciones en problemas multi-modales. Varios métodos han sido propuestos sobre la base de este enfoque [PL06, LW09].

La Tabla 1.1 muestra un resumen de los métodos más representativos de los 4 enfoques principales de los AGNs: *crowding*, *sharing*, *clearing* y *especies*.

Tabla 1.1: Resumen de los métodos basados en nichos

Crowding	*Clearing*
Esquemas de Preselección [Cav70]	*Clearing* [Pét96, Pét97]
Método *Crowding* [DJ75]	Selección por restricción de competencia [LCJ99]
Crowding Determinístico [Mah92]	Selección por restricción de competencia
Selección por Torneo Restringida [Har94]	con búsqueda de patrones [KCJL02]
Crowding Multi-Nicho [CV97]	
Crowding Probabilístico [MG99]	
Sharing	*Especies*
Fitness Sharing [GR87]	Algoritmo Genético para la conservación de especies [LBPC02]
Updating Sharing [CKOS91]	Especies basadas en la optimización de cúmulo de partículas [Li05]
Cluster basado en un esquema *sharing* [YG93]	Especies basadas en Evolución Diferencial [PL06]
Técnicas de identificación de nicho [LW02]	Algoritmo Genético Adaptativo para la conservación de especies [LW09]

1.5. Herramienta KEEL: Algoritmos evolutivos para extraer reglas de asociación disponibles en KEEL

Como hemos visto con anterioridad, en los últimos años muchos autores han propuesto métodos evolutivos para extraer reglas de asociación. Sin embargo, la mayoría no están disponibles para ser usados por cualquier usuario, por lo que su empleo requiere cierta experiencia en programación, y una gran cantidad de esfuerzo y tiempo para escribir el algoritmo que los implementa. Por esta razón hemos integrado un módulo con una gran cantidad de algoritmos para la extracción de reglas de asociación en la herramienta KEEL (*Knowledge Extraction based on Evolutionary Learning*) [AFSG+09].

KEEL es una herramienta software no comercial escrita en Java, la cual permite al usuario emplear algoritmos evolutivos en diferentes tipos de problemas de minería de datos: regresión, clasificación, agrupamiento, etc, incluyendo una gran recopilación de los algoritmos evolutivos existentes para la extracción de reglas de asociación. La Tabla 1.2 resume la lista de estos algoritmos, identificando los algoritmos clásicos basados en la extracción de ítems frecuentes, los algoritmos genéticos mono-objetivos y los AEMOs que han sido diseñados para extraer reglas de asociación. El uso de esta herramienta ofrece varias ventajas:

- Primero, reduce el trabajo de programación. Incluye una librería con algoritmos diseñados para extraer reglas de asociación, especialmente los algoritmos evolutivos para la extracción de RACs. Además, esta herramienta libera a los investigadores del esfuerzo de programación, permitiéndoles centrarse en el análisis de las reglas obtenidas y en la comparación con otros algoritmos ya existentes.

- Segundo, amplía el rango de posibles usuarios de estos algoritmos. Es un software fácil de usar, con una gran cantidad de propuestas ya implementadas, por lo que reduce considerablemente el nivel de conocimientos y experiencia requeridos a la hora de realizar un estudio en esta temática, incluso para investigadores con menor experiencia.

- Tercero, gracias al empleo de un paradigma estricto de orientación a objetos tanto en la librería como en la herramienta software, ambos pueden ser utilizados en cualquier máquina Java, con independencia del sistema operativo existente. Esto simplifica considerablemente el empleo de la herramienta por parte de cualquier investigador.

Tabla 1.2: Algoritmos para extraer RACs disponibles en KEEL

Algoritmos	*Referencia*	*Basado en ítems frequentes*	*AG Mono-Objetivo*	*AEMO*
Apriori	[SA96]	✓		
Eclat	[Zak00]	✓		
GENAR	[MAR01]		✓	
GAR	[MAR02]		✓	
EARMGA	[YZZ09]		✓	
Alatasetal	[AA06]		✓	
MOEA_Ghosh	[GN04]			✓
MODENAR	[AA08]			✓
ARMMGA	[QNMB11]			✓

Por otra parte, KEEL permite ejecutar sus algoritmos dentro del propio entorno (ejecución *on-line*) o generarlos para una ejecución posterior en distintas máquinas (ejecución *off-line*) . La versión actual de KEEL está compuesta por los siguientes módulos (ver Figura 1.5):

Figura 1.5: Pantalla principal de KEEL

- *Tratamiento de datos* (Data Management): Este módulo contiene una serie de herramientas de tratamiento de datos: importación, exportación, edición y visualización de datos, aplicación de transformaciones, etc.

- *Experimentos* (Experiments): Este módulo genera procedimientos de análisis y evaluación automática de algoritmos *off-line*, proporcionando numerosas opciones: tipo de validación, tipo de aprendizaje (clasificación, regresión, aprendizaje no-supervisado), etc.

- *Educacional* (Educational): Este módulo realiza la ejecución de algoritmos *on-line*. Tiene una estructura similar al módulo anterior, pero permite diseñar experimentos para ser ejecutados paso a paso, con propósitos educativos.

- *Módulos* (Modules): En este módulo se puede acceder a varios módulos que amplían KEEL, aquí se incluyen un módulo de aprendizaje no balanceado, un módulo de análisis estadístico no paramétrico, y un módulo de aprendizaje multi-instancia.

- *Ayuda* (Help): Este módulo informa sobre las funcionalidades de KEEL y cómo usar su interfaz gráfica.

La estructura de KEEL sea interesante para distintos tipos de usuarios, dependiendo de sus necesidades. A nivel general, las principales características de KEEL son las siguientes:

- Incluye algoritmos de aprendizaje de modelos predictivos, de preprocesamiento (transformación de datos, discretización, selección de instancias y selección de características) y postprocesamiento, con especial atención a las propuestas evolutivas.

- Permite crear experimentaciones conteniendo múltiples conjuntos de datos y algoritmos conectados entre sí. Los experimentos son generados mediante scripts independientes de la interfaz de usuario, para permitir una ejecución *off-line* en la misma u otras máquinas.

- Posee una librería estadística para analizar resultados de algoritmos. Dicha librería contiene tests estadísticos para analizar la bondad de los resultados obtenidos y también para realizar comparaciones paramétricas y no paramétricas.

- Ofrece al usuario una interfaz amigable, orientada al análisis de algoritmos.

A continuación presentamos un ejemplo de creación de un experimento con la herramienta KEEL. Este ejemplo está centrado en la comparación de dos algoritmos para extraer RACs, un algoritmo genético mono-objetivo y otro AEMO para obtener RACs, EARMGA [YZZ09] y MODENAR [AA08], respectivamente. Para ello utilizamos la BD basketball, la cual está disponible en el repositorio de datos de KEEL [AFFL$^+$11], a través del enlace: http://sci2s.ugr.es/keel/datasets.php. Los valores considerados para los parámetros de entrada de cada método son:

- EARMGA: *Tamaño de fijo de la regla de asociación* = 2, *Tamaño de la población* = 100, *Número de Evaluaciones* = 50000, *Probabilidad de selección* = 0.75, *Probabilidad de cruce* = 0.7, *Probabilidad de mutación* = 0.1, *Número de particiones para atributos numéricos* = 4.
- MODENAR: *Tamaño de la población* = 100, *Número de Evaluaciones* = 50000, *Índice de Cruce* = 0.3, *Umbral* = 60, *Peso soporte* = 0.8, *Peso confianza* = 0.2, *Peso Comprensibilidad* = 0.1, *Peso Amplitud Intervalos* = 0.4.

Para realizar este experimento en KEEL, primero seleccionamos la opción *Experiments* del menú principal de KEEL y definimos el experimento como problema de aprendizaje no supervisado. Después, el primer paso de configuración del grafo del experimento consiste en seleccionar los conjuntos de datos que serán utilizados en el experimento, en este ejemplo seleccionaremos la BD basketball. Luego de seleccionar la BD, se arrastran los métodos EARMGA y MODENAR al espacio de trabajo, y se establecen conexiones entre los métodos y los datos. En cualquier momento, todos los parámetros de los algoritmos pueden ser ajustados haciendo doble click en el nodo del algoritmo. Por otro lado, los arcos que conectan los nodos en KEEL, representan una relación entre ellos, en este caso de intercambio de datos. La Figura 1.6 muestra el grafo del experimento diseñado para este ejemplo y la ventana de configuración del método MODENAR.

Una vez que el grafo ha sido definido, podemos guardar el experimento en un fichero .zip para su ejecución *off-line*. Dicho experimento estará compuesto por un conjunto de scripts XML y un programa .jar que los ejecuta. En el directorio *results* se almacenarán los resultados de cada método durante su ejecución. Para los algoritmos de extracción de reglas de asociación los ficheros de resultados describen la representación de las reglas obtenidas, los valores de las medidas de calidad alcanzados por cada regla y un resumen del promedio de las medidas analizadas para todo el conjunto de reglas.

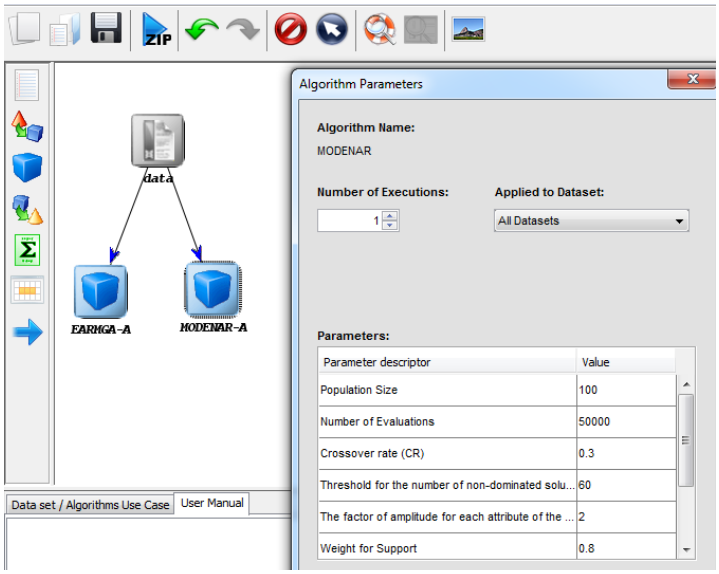

Figura 1.6: Ejemplo de un experimento en KEEL y la ventana de configuración de uno de los métodos

Los resultados obtenidos del análisis de los métodos se muestran en la Tabla 1.3, donde $\#R$ representa el número medio de reglas, Med_{Sop}, Med_{Conf}, Med_{Lift}, Med_{Conv}, Med_{FC}, $Med_{Netconf}$, Med_{YulesQ} representan los valores medios de soporte, confianza, lift, conviction, FC, netconf y yule'sQ, respectivamente, Av_{Amp} el número medio de atributos involucrados en las reglas y $\%Ejem$ es el tanto por ciento de ejemplos de la BD cubiertos por las reglas generadas.

Tabla 1.3: Resultados obtenidos por los métodos analizados en la BD basketball

Algoritmos	$\#R$	Med_{Sop}	Med_{Conf}	Med_{Lift}	Med_{Conv}	Med_{FC}	$Med_{Netconf}$	Med_{YulesQ}	Med_{Amp}	$\%Ejem$
EARMGA	100	0.23	**1**	1.06	∞	0.24	0.03	0.14	2	**100**
MODENAR	57	**0.25**	0.84	**3.72**	∞	**0.43**	**0.28**	**0.38**	2.41	95.84

Analizando los resultados de la Tabla 1.3 podemos destacar que el AEMO MODENAR obtiene un conjunto reducido de reglas, con mejores resultados en

todas las medidas de interés que EARMGA, el cual presenta valores para la mayoría de las medidas cercanos al valor de independencia que estas medidas pueden tomar (ver sección 1.2.1). Destacar que ambos algoritmos obtienen buenos valores de cubrimiento para esta BD.

Capítulo 2

Nuevo Algoritmo Evolutivo Multi-Objetivo para Extraer Reglas de Asociación Cuantitativas

Como hemos visto en el capítulo 1, la extracción de reglas de asociación ha sido planteada por algunos investigadores como un problema multi-objetivo, donde se traten de optimizar varias funciones objetivos a la vez, ofreciendo al usuario un conjunto de reglas de mayor calidad en función de varias medidas. Los AEMOs han probado ser un mecanismo interesante para tratar con problemas de naturaleza multi-objetivo, proporcionando una familia de soluciones igualmente validas, donde cada solución tiende a satisfacer un criterio de la búsqueda en mayor medida que cualquier otra solución. Debido a ello, en los últimos años algunos investigadores han propuesto AEMOs para extraer reglas de asociación, donde cada regla representa una solución del frente de Pareto (ver subsección 1.4.2 del capítulo 1).

En este capítulo proponemos QAR-CIP-NSGA-II, un modelo evolutivo multi-objetivo que extiende el ampliamente utilizado AEMO NSGA-II [DAPM02] para

extraer RACs de muy buena calidad, con un buen equilibrio entre las diferentes medidas de interés y un buen cubrimiento de la BD. Para lograr esto, nuestra propuesta maximiza tres objetivos: comprensibilidad, interés y rendimiento, realizando un aprendizaje evolutivo de los intervalos de los atributos y una selección de las condiciones para cada regla. Además, nuestro modelo introduce dos nuevos componentes al modelo evolutivo de NSGA-II: un proceso de reinicialización y una población externa (PE), con el fin de promover diversidad en la población, almacenar todas las reglas no dominadas encontradas y mejorar el cubrimiento de la BD. Esta propuesta presenta un enfoque independiente de la BD porque no necesita definir umbrales mínimos para el soporte y la confianza, los cuales son difíciles de determinar para cada BD.

Este capítulo se organiza como sigue. En la siguiente sección se detalla el algoritmo evolutivo propuesto para obtener RACs. En la sección 2.2 presentamos los resultados del estudio experimental realizado sobre 26 BDs reales para evaluar la efectividad del método propuesto. Finalmente, en la sección 2.3 se presenta un breve resumen del capítulo.

2.1. QAR-CIP-NSGA-II: Algoritmo Evolutivo para extraer reglas de asociación cuantitativas

En esta sección se describe nuestra propuesta para obtener un conjunto de reglas de asociación de alta calidad en función de varias medidas de interés. Este modelo utiliza el AEMO NSGA-II [DAPM02] para realizar un aprendizaje evolutivo de las reglas e introduce dos nuevos componentes a este modelo evolutivo: una PE y un proceso de reinicialización. En las siguientes subsecciones describiremos en detalle cada uno de las características de esta propuesta.

2.1.1. Dos nuevos componentes en el modelo evolutivo: PE y proceso de reinicialización

Esta propuesta extiende el conocido AEMO NSGA-II [DAPM02] e introduce a este modelo una PE y un proceso de reinicialización para almacenar todas las reglas no dominadas encontradas, promover diversidad en la población, y mejorar el cubrimiento de la BD. La PE almacenará todas las reglas no dominadas encontradas y se actualizará al final de cada generación con las reglas no dominadas de

la población actual. Las reglas no dominadas que sean redundantes se eliminarán de la PE para evitar solapamiento entre las reglas. Una regla la consideraremos redundante si los intervalos de todos sus atributos están contenidos dentro de los intervalos de los atributos de otra regla. Destacar que, el tamaño de la PE no está limitado, lo que nos permite devolver un conjunto de reglas no dominadas independientemente del tamaño de la población y reducir el tamaño de la población (independientemente de la magnitud del problema), lo cual ayuda a controlar mejor la convergencia del método. Por otro lado, aunque el tamaño de la PE no está limitado, por lo general el conjunto reducido de reglas será reducido, porque los criterios de no dominancia nos permite mantener sólo las reglas del frente de Pareto y las reglas redundantes son eliminadas.

Además, para evitar caer en los óptimos locales y provocar diversidad en la población aplicamos un proceso de reinicialización cuando el número de nuevos individuos de la población en una generación es menor que un $\alpha\,\%$ del tamaño de la población actual (α definido por el usuario, normalmente al $5\,\%$). Este proceso marca los ejemplos que hayan sido cubiertos por las reglas de la PE y aplica nuevamente el proceso de inicialización sobre los ejemplos que no estén marcados, lo cual nos permite realizar una buena exploración del espacio de búsqueda. Por último, se actualiza la PE con la población nueva siguiendo el criterio de no dominancia.

Téngase en cuenta que ambos componentes son complementarios. El proceso de reinicialización utiliza los ejemplos no cubiertos por las reglas de la PE para generar la nueva población. Por otro lado, la PE mantiene todas las reglas no dominadas encontradas hasta el último momento, evitando que las soluciones no dominadas sean eliminadas cuando el proceso de reinicialización reinicia toda la población.

2.1.2. Esquema de Codificación y Población Inicial

Un cromosoma es un vector de n genes que representa los atributos e intervalos de una regla, donde n es el número de atributos de la BD. Nuestra propuesta utiliza una codificación posicional en la que el i-ésimo gen codifica el i-ésimo atributo. Para combinar la selección de las condiciones con el aprendizaje de los intervalos, cada gen consta de cuatro partes:

- La primera parte (ac) indica si un gen es considerado en la regla. Cuando esta parte es '-1', este atributo no está involucrado en la regla, y cuando esta

parte es '0' o '1' este atributo es parte del antecedente o del consecuente de la regla, respectivamente. Todos los genes que tengan '0' en su primera parte formarán parte del antecedente de la regla mientras los genes que tengan '1' formarán parte del consecuente de la regla.

- La segunda parte representa el límite inferior (li) del intervalo del atributo.
- La tercera parte representa el límite superior (ls) del intervalo del atributo

Si el atributo es nominal, li y ls serán iguales, representando solo un valor del atributo nominal. Por tanto un cromosoma C_T se codifica de la siguiente manera, donde n es el número de atributos en la BD.

$$C_T = Gen_1 Gen_2 \dots Gen_n, \;\; i = 1, \dots, n$$
$$Gen_i = (ac_i, li_i, ls_i)$$

I

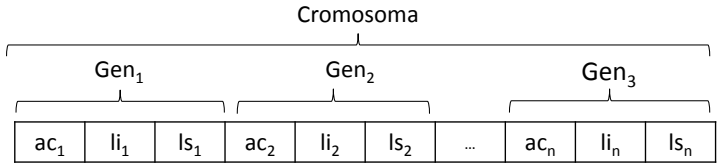

Figura 2.1: Esquema de un cromosoma codificado por QAR-CIP-NSGA-II

Para evitar que el aumento de los intervalos cubra la totalidad del dominio hemos definido *Amplitud*, el cual representa el tamaño máximo que el intervalo de un atributo puede alcanzar. La *Amplitud* de un atributo i se define como:

$$Amplitud_i = (Max_i - Min_i)/\gamma \tag{2.1}$$

donde γ es un valor dado por el experto que nos permite determinar el equilibrio entre la generalización y la especificidad de las reglas, y Min_i y Max_i son los valores de mínimo y máximo del dominio del atributo i, respectivamente.

La población inicial estará compuesta por un conjunto de reglas que presentan un buen cubrimiento de la BD y contienen un solo atributo en el consecuente, aunque este esquema de codificación nos permite codificar reglas con más de una atributo en el consecuente. Para crear la población inicial, primero se selecciona

aleatoriamente los atributos que formarán parte del antecedente y del consecuente de la regla (al menos seleccionaremos un atributo para el antecedente y el consecuente). Luego se selecciona aleatoriamente un ejemplo de la BD, el cual será utilizado para generar los intervalos de cada atributo de la regla. Para crear un intervalo se generan sus límites de manera que el valor del ejemplo seleccionado quede en el centro del intervalo. Además, este intervalo tendrá un tamaño igual al 50 % del valor de *Amplitud* del atributo. Si algún límite del intervalo supera el límite del dominio de su atributo, entonces se reemplaza el límite del intervalo por el límite del dominio del atributo. Finalmente se marcan los ejemplos de la BD que cubre la regla para generar las próximas reglas a partir de los ejemplos que no han sido marcados, de esta manera garantizamos obtener una población inicial con un buen cubrimiento de la BD. Este proceso se repite hasta que se complete la población inicial. Si se han marcado todos los ejemplos de la BD y la población inicial no se ha completado, se vuelven a desmarcar todos los ejemplos y el proceso se repite hasta que la población inicial sea completada. Por último, se inicializa la PE con la reglas no dominadas de la población inicial.

La Figura 2.2 representa un esquema del proceso de creación de la población inicial.

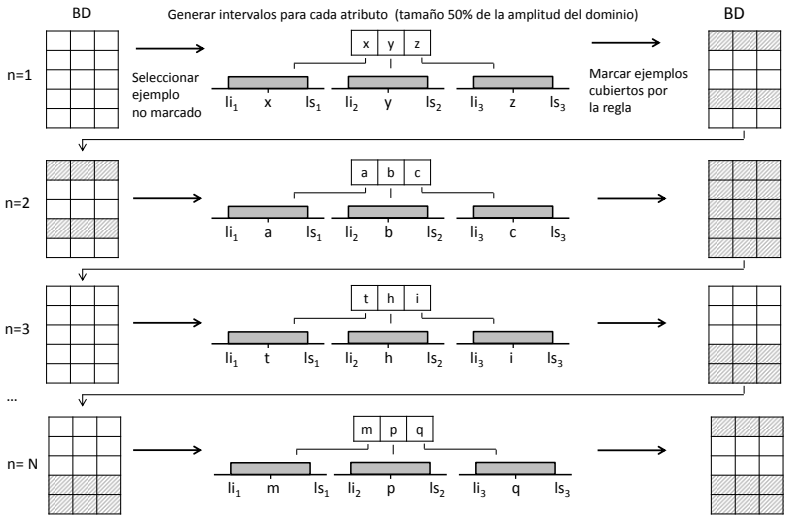

Figura 2.2: Esquema del proceso de inicialización de la población con un tamaño de N individuos

Tabla 2.1: Seis ejemplos para la BD de un ejemplo sobre el cálculo de los intervalos de los atributos de una regla

ID	X_1	X_2	X_3
ID1	0,3	23	3,2
ID2	1,0	38	20,5
ID3	0.9	12	10,1
ID4	0,0	20	50,3
ID5	0,4	50	70,8
ID6	0,2	10	5,9
Límite inferior del dominio	0,0	10	3,2
Límite superior del dominio	1,0	50	70,8
50 % de la $Amplitud_i$	0,25	10	16,9

Por ejemplo, consideremos una BD simple con tres atributos X_1, X_2 y X_3, 6 ejemplos y $\delta = 2$. La Tabla 2.1 muestra los 6 ejemplos de la BD, el límite inferior y superior del dominio de los atributos y el 50 % de la $Amplitud_i$ de cada atributo. Supongamos que seleccionamos al azar los atributos X_1 y X_2 para el antecedente y consecuente de la regla, respectivamente y el registro ID3. En esta iteración del proceso de inicialización se genera la regla $X_1 \in [0‘77,1‘0] \rightarrow X_2 \in [10,17]$, calculando sus intervalos de la siguiente manera:

$$li_1 = max\left\{0‘9 - \frac{0‘25}{2}, 0‘0\right\} = 0,77 \qquad ls_1 = min\left\{0‘9 + \frac{0‘25}{2}, 1‘0\right\} = 1,0$$

$$li_2 = max\left\{12 - \frac{10}{2}, 10\right\} = 10 \qquad ls_2 = min\left\{12 + \frac{10}{2}, 50\right\} = 17$$

$$li_3 = max\left\{10‘1 - \frac{16‘9}{2}, 3‘2\right\} = 3,2 \qquad ls_3 = min\left\{10‘1 + \frac{16‘9}{2}, 70‘8\right\} = 18,55$$

Para que cada valor del ejemplo ID3 quede en el centro del intervalo le sumamos (límite superior ls_i) o le restamos (límite inferior li_i) la mitad del 50 % de la *Amplitud* del intervalo correspondiente. Si en algún caso se sobrepasan los límites del dominio de algún atributo, se sustituye el límite del intervalo por el mismo límite del dominio del atributo correspondiente. Por ejemplo, el ls_1 es 1,0 porque el límite del intervalo superó el límite superior del dominio del atributo.

La Figura 2.3 muestra el cromosoma generado en este ejemplo. Esta regla cubre los ejemplos ID1 e ID3 como podemos ver en la Tabla 2.1. En esta situación, est os no ma

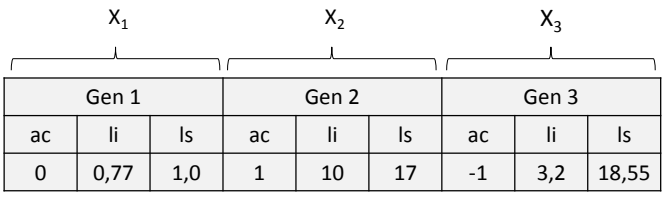

Figura 2.3: Cromosoma obtenido para el ejemplo del cálculo de los intervalos de los atributos de una regla

Finalmente, la PE se inicializa con las reglas no dominadas de la población inicial.

2.1.3. Objetivos

Nuestra propuesta maximiza tres objetivos: rendimiento, interés y comprensibilidad. El rendimiento es definido como el producto entre el soporte y el FC (ver sección 1.2.1 del capítulo 1). Este objetivo nos permite dirigir la búsqueda hacia un conjunto de reglas con un buen equilibrio entre reglas específicas y generales. Destacar que esta propuesta solo genera reglas con dependencia fuerte [BBSV02] entre los ítems porque representan dependencias positivas entre ellos y evitan el problema del soporte (ver sección 1.2.2 del capítulo 1). Para ello, toda regla $X \rightarrow Y$ debe cumplir:

- $FC(X \rightarrow Y) > 0$
- $Soporte(X \rightarrow Y) >$ mínimo de soporte
- $\neg(Soporte(X \rightarrow Y) >$ (1 - mínimo de soporte))

Esto hace que la medida de rendimiento tome valores en el intervalo [0, 1], donde resultan más útiles para el usuario las reglas con un valor de rendimiento cerca de 1.

El *interés* mide como de interesante es una regla con el fin de extraer solo aquellas reglas interesantes para el usuario. En esta propuesta hemos utilizado la medida de interés lift (ver sección 1.2.1 del capítulo 1), la cual nos permite detectar dependencias negativas, positivas o independencia entre los ítems. Además, como su rango de valores no está limitado, permite diferenciar más las reglas y reducir el número de empates.

La *comprensibilidad* mide lo fácil de interpretar que puede ser una regla [FLF00]. Para los usuarios las reglas que involucran muchos atributos son más difíciles de comprender. En este trabajo hemos usado la medida de comprensibilidad de una regla $X \rightarrow Y$ según el número de atributos que contiene. Esta se define como sigue, donde $Atrib_{X \rightarrow Y}$ es el número de atributos involucrados en el antecedente de la regla.

$$Comprensibilidad(X \rightarrow Y) = 1/Atrib_{X \rightarrow Y} \tag{2.2}$$

2.1.4. Operadores

El operador de cruce genera dos hijos intercambiando aleatoriamente los genes de dos [illegible] ejemplo sencill

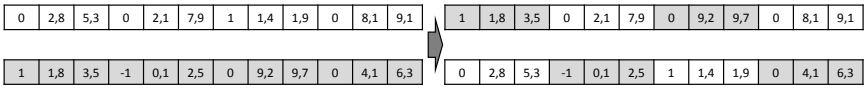

Figura 2.4: Un ejemplo simple del operador de cruce

El operador de mutación selecciona aleatoriamente un gen del cromosoma. De este gen, modifica el valor de *ac* aleatoriamente dentro del conjunto {-1,0,1}. Después, selecciona al azar uno de los límites del intervalo y aumenta o disminuye su valor de manera aleatoria. Destacar que la forma en que modificamos el intervalo es similar a la calculada en el proceso de inicialización, teniendo en cuenta que no debemos superar el tamaño de la amplitud del intervalo. Para disminuir el intervalo se seleccionará un nuevo valor aleatoriamente entre el valor actual del límite y el valor del otro límite del intervalo y si se va a aumentar el intervalo, entonces el valor será seleccionado entre el valor actual del límite y el límite del dominio del atributo.

Después de aplicar el operador de mutación, se aplica el operador de reparación

para corregir las reglas que tengan más de un atributo en el consecuente o que no tengan ningún atributo en el antecedente o consecuente. Si el consecuente contiene más de un atributo, uno de ellos se selecciona aleatoriamente para ser el consecuente y el resto pasan al antecedente. Si no hay ningún atributo en el antecedente y/o consecuente, se seleccionan al azar entre los atributos que no han sido considerados en la regla. Además para obtener reglas más simples este operador decrementa el tamaño de los intervalos mientras que los ejemplos cubiertos sean los mismos que los ejemplos cubiertos por los intervalos originales.

Observe que hemos utilizado operadores genéticos comunes que funcionan bien para extraer reglas de asociación en vez de los operadores genéticos del modelo NSGA-II diseñados para problemas de optimización multi-objetivo.

2.1.5. Modelo Evolutivo Multi-Objetivo

Con las modificaciones comentadas anteriormente, el modelo evolutivo sería el siguiente. En primer lugar nuestra propuesta genera una población inicial e inicializa la PE con las reglas no dominadas de la población inicial. Luego, se genera una población de hijos a partir de la población actual aplicando los operadores de selección, cruce y mutación. La próxima población se construye con los mejores individuos de la población resultando de unir la población actual y la población de hijos, se actualiza la PE con la población actual y, si es necesario, se aplica el proceso de reinicialización. Todo este proceso se repite hasta que se cumpla la condición de parada. El algoritmo NSGA-II tiene dos características que lo convierten en un paradigma dentro de los AEMOs. La primera es la evaluación de cada solución basada en el ranking de Pareto y en el operador de "crowding", y la segunda es el proceso elitista que propone para actualizar la población en cada generación.

Cada solución en la población actual es evaluada de la siguiente manera. En primer lugar, se le asigna rango 1 a todas las soluciones no dominadas en la población actual, las cuales son tentativamente eliminadas de la población actual. Luego, se le asigna rango 2 a todas las soluciones no dominadas en la población actual reducida, las cuales también son tentativamente eliminadas de la población actual. Este procedimiento se realiza hasta que todas las soluciones hayan sido tentativamente eliminadas de la población actual, es decir, hasta que se haya asignado un rango a todas las soluciones. Como resultado, se ha asignado un rango diferente a cada solución, donde las soluciones con rangos más pequeños se consideran mejores que las de rangos más altos. Para las soluciones del mismo

rango, se tiene en cuenta un criterio adicional llamado medida de "crowding".

Para determinar la medida de "crowding" de una solución se calcula la distancia entre las soluciones adyacentes con su mismo rango en el espacio de los objetivos. Las soluciones en espacios menos poblados con mayores valores de la medida de "crowding" se consideran mejores que las soluciones más pobladas con menores valores para esta medida.

La selección de un par de padres de la población actual se realiza utilizando selección por torneo binario basada en el ranking de Pareto y la medida de "crowding". Para construir la próxima población, se combinan la población actual y la población de hijos en una población mixta. Cada solución de la población mixta es evaluada de la misma forma que en el proceso de selección de los padres, utilizando el ranking de Pareto y la medida de "crowding". La próxima población se construye eligiendo un número específico (el tamaño de la población) de las mejores soluciones de la población mixta.

La Figura 2.5 presenta un esquema del método evolutivo.

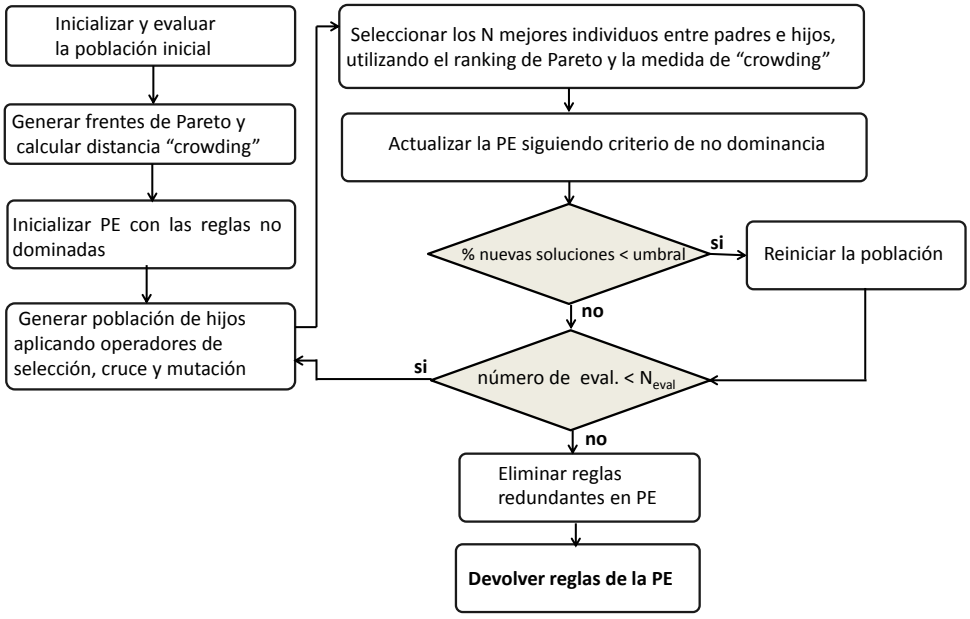

Figura 2.5: Organigrama del método QAR-CIP-NSGA-II

2.1.6. Pasos del algoritmo

De acuerdo con la descripción anterior, el algoritmo propuesto para extraer RACs se puede resumir en los siguientes pasos:

Entrada: tamaño de la población N, número de evaluaciones N_{eval}, probabilidad de mutación P_{mut}, factor de amplitud para cada atributo de la BD δ, umbral de diferencia α.

Salida: Reglas de la PE

Paso 1: Inicialización

a) Generar la población inicial (P_0) con N cromosomas.

b) Evaluar la población inicial.

c) Generar todos los frentes no dominados $F = (F_1, F_2, ...)$ de la población inicial y calcular la distancia "crowding" para F_i.

d) Inicializar la PE.

Paso 2: Generar la población de hijos (Q_t) de la siguiente manera:

a) Seleccionar un par de padres utilizando selección por torneo binario basada en el ranking de Pareto y para las soluciones del mismo ranking se utiliza la medida de "crowding".

b) Sobre los dos padres seleccionados, se aplica el operador de cruce, este operador genera dos hijos a partir de intercambiar los genes de los dos padres. A continuación, los operadores de mutación y reparación se aplican para los dos hijos.

c) Evaluar los nuevos individuos. Este proceso se repite hasta que se complete la población de hijos.

Paso 3: Generar la siguiente población (P_{t+1}) de la siguiente manera:

a) Crear la población R_t uniendo las poblaciones P_t y Q_t.

b) Generar todos los frentes no dominados $F = (F_1, F_2, ...)$ de la población R_t y calcular la distancia "crowding" para F_i.

c) Crear P_{t+1} con los mejores cromosomas de R_t utilizando el ranking de Pareto y la distancia "crowding":

- Incluir el i-ésimo frente no dominado en P_{t+1}.
- Verificar el próximo frente para su inclusión.
- Ordenar descendentemente utilizando el operador 'crowding".
- Seleccionar los primeros $(N - |P_{t+1}|)$ elementos de F_i.

Paso 4: Actualización de la PE, siguiendo el criterio de no dominancia.

Paso 5: Si la diferencia entre la población actual y la población anterior es menor que $\alpha\,\%$, entonces se reinicia la población.

Paso 6: Si no se ha alcanzado el número máximo de evaluaciones, ir al Paso 2.

Paso 7: Eliminar la redundancia en la PE, eliminando los cromosomas que son subcromosomas de otros. Un subcromosoma es un individuo en el que los intervalos de todos sus genes están contenidos dentro de los intervalos de los genes de otro cromosoma.

Paso 8: Devolver las reglas de la PE.

2.2. Estudio Experimental

Varios experimentos han sido realizados para analizar el funcionamiento de nuestra propuesta sobre 26 BDs. Esta sección se organiza como sigue:

- En la subsección 2.2.1 se presenta una descripción del marco de experimentación que incluye: las BDs que se utilizan en estos experimentos y la configuración de los parámetros de los métodos considerados para la comparación.

- En la subsección 2.2.2 se compara nuestro enfoque con el modelo evolutivo original NSGA-II para analizar la influencia de los nuevos componentes introducidos.

- En la subsección 2.2.3 se compara nuestro método con cuatro algoritmos genéticos mono-objetivos (EARMGA [YZZ09], GAR [MAR02], GENAR [MAR01] y Alatasetal [AA06]) y tres AEMOs (ARMMGA [QNMB11], MODENAR [AA08] y MOEA_Ghosh [GN04]) para extraer RACs.

Tabla 2.2: BDs consideradas en el estudio experimental

Nombres	#Atrib(R/E/N)	#Ejem	Nombres	#Atrib(R/E/N)	#Ejem
Balance Scale (ba)	5 (5/0/0)	625	Satimage (sa)	37 (0/37/0)	6435
Basketball (bas)	5 (3/2/0)	96	Segment (se)	20 (19/1/0)	2310
Bolts (bo)	8 (2/6/0)	40	Sonar (so)	61 (60/0/1)	208
Coil2000 (co)	86 (0/86/0)	9822	Spambase (sp)	58 (57/1/0)	4597
House_16H (hh) [1]	17(10/7/0)	22784	Spectfheart (spe)	45 (0/45/0)	267
Ionosphere (io)	34 (32/1/1)	351	Stock Price (st)	10 (10/0/0)	950
Letter (le)	16 (0/16/0)	20000	Stulong (stu)[2]	5(5/0/0)	1419
Magic (ma)	11(10/0/1)	19020	Texture (te)	41 (40/1/0)	5500
Movement Libras (mo)	91 (90/0/1)	360	Thyroid (th)	22 (6/16/0)	7200
Optdigits (op)	65 (0/65/0)	5620	Vehicle (ve)	19 (0/18/1)	846
Penbased (pe)	16 (0/16/0)	10992	Wdbc (wd)	31 (30/0/1)	569
Pollution (po)	16 (16/0/0)	60	Wine (wi)	14 (13/1/0)	178
Quake (qu)	4 (3/1/0)	2178	Vowel (vo)	14 (10/4/0)	990

Disponible en `http://sci2s.ugr.es/keel/datasets.php`

- En la subsección 2.2.4 se compara nuestro método con dos algoritmos clásicos de extracción de reglas de asociación: Apriori [Bor03, SA96] y Eclat [Zak00].
- Finalmente, en la subsección 2.2.5 se analiza la escalabilidad de nuestra propuesta.

2.2.1. Experimentos

En estos experimentos hemos seleccionado 26 BDs del mundo real con distintos tamaños para analizar la efectividad de nuestra propuesta. La Tabla 2.2 resume las principales características de las 26 BDs y muestra el enlace al repositorio de datos KEEL-dataset [AFFL$^+$11] del cual podemos descargarlas. Para cada BD se muestra el número de ejemplos ("#Ejem") y el número de atributos reales, enteros o nominales que contiene ("#Atrib(R/E/N)"). Para desarrollar los diferentes experimentos, consideramos los resultados promedio de 5 ejecuciones sobre cada BD.

[1] Esta BD fue diseñada sobre la base de los datos proporcionada por la Oficina del Censo de EE.UU. [http://www.census.gov] (Acceso Lookup [http://www.census.gov/cdrom/lookup]: Resumen del archivo 1).

[2] Este estudio se realizó en el 2do Departamento de Medicina, de la 1ra Facultad de Medicina

En el estudio experimental hemos utilizado los algoritmos genéticos mono-objetivo EARMGA [YZZ09], GAR [MAR02], GENAR [MAR01] y Alatasetal [AA06] y los AEMOs ARMMGA [QNMB11], MODENAR [AA08] y MOEA_Ghosh [GN04] (ver subsección 1.4.1 y 1.4.2 del capítulo 1, respectivamente). Además, comparamos nuestra propuesta con los algoritmos clásicos Apriori [Bor03, SA96] y Eclat [Zak00] (ver subsección 1.3 del capítulo 1). Los parámetros utilizados para estos métodos son mostrados en la Tabla 2.3. Con estos parámetros para nuestra propuesta, hemos tratado de facilitar las comparaciones, seleccionando parámetros estándar comunes que funcionan bien en la mayoría de los casos, y para el resto de los algoritmos los hemos seleccionado de acuerdo a las recomendaciones de los autores de cada propuesta, los cuales son los parámetros por defecto incluidos en la herramienta KEEL [AFSG$^+$09].

Destacar que Apriori, Eclat y GAR necesitan un mínimo de soporte y confianza para extraer RACs, seleccionando para ellos valores estándar que funcionan bien en la mayoría de los casos para todas las BDs. Además los resultados obtenidos para los AEMOs se refieren al conjunto de reglas no dominadas obtenido.

2.2.2. Análisis de los nuevos componentes introducidos en el algoritmo evolutivo multi-objetivo

En esta sección se estudia el rendimiento de nuestra propuesta en comparación con el enfoque clásico NSGA-II con el fin de analizar el rendimiento de los nuevos componentes introducidos en el modelo evolutivo NSGA-II: la PE y el proceso de reinicialización. Para ello, hemos extendido el enfoque clásico (lo llamaremos QAR-CIP-NSGA-II_C) para extraer reglas de asociación, utilizando el mismo esquema de codificación, objetivos, población inicial y operadores genéticos que en nuestra propuesta.

La Tabla 2.4 muestra los resultados medios obtenidos por los métodos en todas las BD, donde $\#R$ representa el número medio de reglas, Med_{Sop}, Med_{Conf},

de la Universidad Carolina y el Hospital de la Universidad Carolina, bajo la supervisión del Prof. F. Boudk con la colaboración de M. Tomeckov y Ass. Prof. J. Bultas. Los datos se transfirieron al formato electrónico por el Centro Europeo de Informática Médica, Estadística y Epidemiología de la Universidad Carolina y la Academia de Ciencias. La fuente de datos está disponible en la página web http://euromise.vse.cz/challenge2004. En la actualidad, el análisis de los datos es soportado bajo la subvención del Ministerio de Educación CR Nr LN 00B 107.

Tabla 2.3: Parámetros considerados en la comparación de los métodos

Algoritmos	*Parámetros*
Apriori	*mínimo de soporte = 0,1, mínimo de confianza = 0,8*
Eclat	*mínimo de soporte = 0,1, mínimo de confianza = 0,8*
Alatasetal	N_{eval}=50000, *nCromoInicialAleat=12,* $r = 3$, *TamTorneo = 10,* P_{sel}= 0,25, P_{cru} = 0,7, P_{mut_min} = 0,05, P_{mut_max} = 0,9, $Peso_{sop}$ = 5, $Peso_{conf}$ = 20, $Peso_{amplRule}$ = 0,05, $Peso_{amplInterv}$ = 0,02, $Peso_{cubrimiento}$ = 0,01
EARMGA	*TamPop = 100,* N_{eval} = 50000, $k = 2$, P_{sel} = 0,75, P_{cru} = 0,7, P_{mut} = 0,1, α = 0,01
GENAR	*TamPop = 100,* N_{eval} = 50000, P_{sel}= 0,25, P_{cru} = 0,7, P_{mut} = 0,1, *nReglas = 30, FP = 0,7, AF = 0,2*
GAR	*TamPop = 100, nItems = 100,* N_{eval} = 50000, P_{sel}= 0,25, P_{cru} = 0,7, P_{mut} = 0,1, ω = 0,4, Ψ= 0,7, μ= 0,5, *mínimo de soporte = 0,1, mínimo de confianza = 0,8*
ARMMGA	*TamPop=100,* N_{eval}=50000, *k=2 (3 with HH),* P_{sel}=0,95, P_{cru}=0,85, P_{mut} = 0,01, *db=0,01*
MODENAR	*TamPop = 100,* N_{eval}=50000, *Umbral= 60, CR = 0,3,* $Peso_{sop}$ = 0,8, $Peso_{conf}$ = 0,2, $Peso_{comp}$ = 0,1, $Peso_{amplInterv}$ = 0,4
MOEA_Ghosh	*TamPop = 100,* N_{eval}=50000, *PuntoCruce=2,* P_{cru}=0,8, P_{mut}= 0,02
QAR-CIP-NSGA-II	*TamPop = 100,* N_{eval}=50000, P_{mut}= 0,1, γ=2, α = 5

Tabla 2.4: Resultados del valor medio de las medidas para todas las BDs en la comparación entre QAR-CIP-NSGA-II y el clásico NSGA-II

Algoritmos	$\#R$	Med_{Sop}	Med_{Conf}	Med_{Lift}	Med_{Conv}	Med_{FC}	$Med_{Netconf}$	$Med_{Yule'sQ}$	Med_{Amp}	$Med_{HiperVol}$	$\%Ejem$
QAR-CIP-NSGA-II_C	81,44	**0,29**	0,9	71,79	∞	0,8	0,67	0,91	3,06	4562,87	88,66
QAR-CIP-NSGA-II	**147,26**	0,13	**0,93**	**440,69**	∞	**0,91**	**0,78**	**0,92**	**2,9**	**5098,30**	**92,66**

Med_{Lift}, Med_{Conv}, Med_{FC}, $Med_{Netconf}$, Med_{YulesQ} representan los valores medios de soporte, confianza, lift, conviction, FC, netconf y yule'sQ, respectivamente, Av_{Amp} el número medio de atributos involucrados en las reglas, $Med_{HiperVol}$ representa el valor medio de la medida hipervolumen [EZdF03] y $\%Ejem$ es el tanto por ciento de ejemplos de la BD cubiertos por las reglas generadas. Además, en la Tabla C.1 de la sección C.1 del apéndice C se pueden encontrar los resultados obtenidos por cada método en cada BD. Los valores de la medida hipervolumen se calcularon utilizando el paquete emoa [3] [Mer13] de la herramienta R, el cual implementa el algoritmo propuesto por Fonseca y colaboradores en [CMFLI06].

[3] paquete emoa está disponible en la Red Integral de Archivo R (CRAN) en `http://cran.r-project.org/web/packages/emoa/`

Los valores ∞ que se muestra en la tabla representan el valor máximo para estas medidas (ver subsección 1.2.1 del capítulo 1).

A partir del análisis de los resultados presentados en estas tablas podemos presentar las siguientes conclusiones:

- La PE nos permite obtener un mayor número de reglas no-dominadas del frente de Pareto debido a que el número de reglas no está limitado por el tamaño de la población actual, ofreciendo cada regla un conocimiento interesante sobre la BD.

- El proceso de reinicialización, junto con la PE nos permite realizar una buena exploración del espacio de búsqueda, mejorando el cubrimiento de las BDs.

- QAR-CIP-NSGA-II presenta valores más altos de hipervolumen que el enfoque clásico, obteniendo una mayor área no dominada. Observe que estos valores son muy altos debido a que el rango de la medida lift no está limitado.

- Las reglas obtenidas por nuestra propuesta presenta mejoras en casi todas las medidas de interés y un cubrimiento similar o superior en todas las BDs, lo cual muestra una sinergia positiva entre los nuevos componentes.

Para evaluar si existen diferencias significativas entre los resultados, realizamos un análisis estadístico [GH08, GFLH09, GMLH09], en particular aplicamos tests no paramétricos de acuerdo con las recomendaciones realizadas en [Dem06]. Decidimos aplicar los tests estadísticos a los resultados medios obtenidos por las medidas de interés lift, FC, netconf y yule'sQ. Destacar que no hemos utilizado la medida conviction porque los algoritmos obtienen infinito en la mayoría de las BDs. Para comparar los dos algoritmos utilizamos el test de Wilcoxon [She03, Wil45]. Este test se basa en el cálculo de las diferencias entre dos medias de una muestra (típicamente, significa errores de pruebas obtenidos por un par de algoritmos diferentes en diferentes BDs). En el marco de la clasificación estas diferencias están bien definidas, ya que estos errores están en el mismo dominio. En nuestro caso, para establecer bien las diferencias entre las medidas de interés, proponemos la transformación de sus valores medios a $MediaS$, la cual se define para cada medida como:

Tabla 2.5: Resultados del test de Wilcoxon ($\alpha = 0.05$) en la comparación con el clásico NSGA-II

Medidas de Interés	**Comparación**	R^+	R^-	**Hipótesis**	p-**valor**
FC	QAR-CIP-NSGA-II vs. QAR-CIP-NSGA-II_C	351.0	0.0	**Rechazada**	<0.0001
Netconf	QAR-CIP-NSGA-II vs. QAR-CIP-NSGA-II_C	322.5	28.5	**Rechazada**	<0.0001
Yule'sQ	QAR-CIP-NSGA-II vs. QAR-CIP-NSGA-II_C	202	149	**No Rechazada**	≥ 0.2
Lift	QAR-CIP-NSGA-II vs. QAR-CIP-NSGA-II_C	340.0	11.0	**Rechazada**	<0.0001

- *Para las medidas FC, netconf y yule'sQ:*

$$MediaS = \begin{cases} \frac{|valorMedio|}{2} & si\ valorMedio \leq 0 \\ \\ \frac{|valorMedio|}{2} + 0{,}5 & sino \end{cases}$$

- *Para la medida lift:*

$$MediaS = \begin{cases} 1 - \frac{0{,}5}{valorMedio} & si\ valorMedio > 1 \\ \\ 0{,}5 - \frac{|valorMedio|}{2} & si\ 0 \leq valorMedio \leq 1 \end{cases}$$

donde $valorMedio$ representa el valor medio obtenido para cada medida en una BD. $ValorMedio$ obtiene valores en [0,1], donde el peor valor representa el valor de independencia de cada medida (ver sección 1.2.1 del capítulo 1), ya que no proporciona nuevos conocimientos para el usuario.

La Tabla 2.5 muestra los resultados del test de Wilcoxon para las cuatro medidas. La hipótesis de igualdad para el test de Wilcoxon es rechazada en todos los casos con un p-valor muy pequeño, excepto en la medida yule'sQ, que aunque no obtenemos diferencias significativas, nuestra propuesta alcanza un mayor número de rankings positivos.

La Figura 2.6 muestra los frentes de Pareto obtenidos por nuestra propuesta cuando se aplica el proceso de reinicialización en diferentes momentos del proceso evolutivo en dos BDs (visualiza las soluciones de una sola ejecución). En esta figura, representamos las soluciones de QAR-CIP-NSGA-II en 3 dimensiones y graficamos las proyecciones de estas soluciones en todos los planos posibles de los objetivos. Observe que hemos modificado los objetivos para mostrarlos como

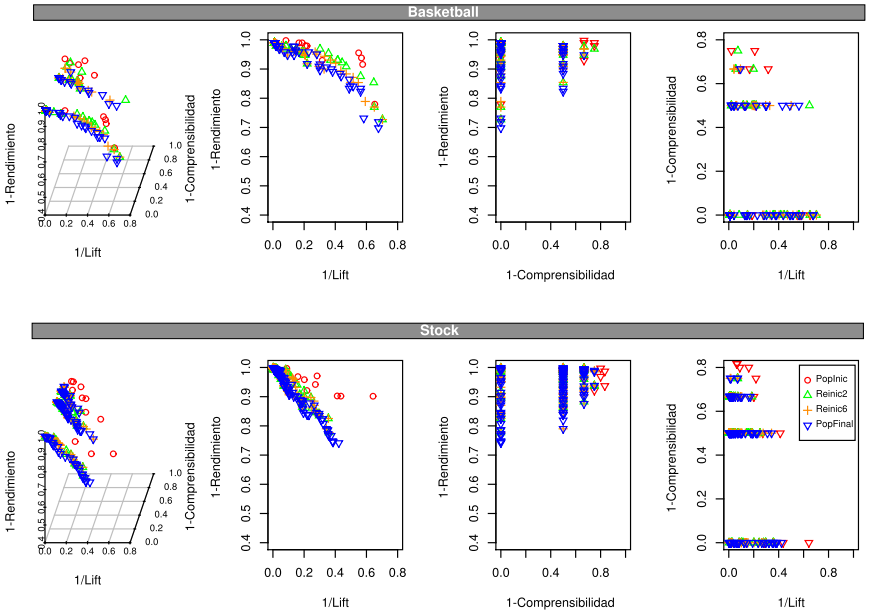

Figura 2.6: Frentes de Pareto obtenidos en diferentes momentos del proceso evolutivo en dos BDs

objetivos de minimización. Además para mantener toda la información, no hemos eliminado las soluciones dominadas que se obtienen en las proyecciones. Podemos ver cómo mejora el frente de Pareto cuando aumenta el número de veces que se aplica el proceso de reinicialización. Por otra parte, se puede ver fácilmente en estas figuras cómo el proceso de reinicialización y la PE nos permiten mejorar los frente Pareto y aumentar el número de soluciones no dominadas para cada proceso de reinicialización.

2.2.3. Comparación con otros enfoques evolutivos mono-objetivos y multi-objetivos

En esta sección se analiza el rendimiento de nuestro algoritmo en comparación con cuatro algoritmos mono-objetivo y tres AEMOs para extraer RACs. La Tabla 2.6 muestra los resultados medios obtenidos por los métodos en todas las BD (la cabecera de esta tabla ha sido introducida en la sección anterior). Los resultados

Tabla 2.6: Resultados del valor medio de las medidas para todas las BDs en la comparación entre QAR-CIP-NSGA-II y los métodos evolutivos

Algoritmos	*#R*	Med_{Sop}	Med_{Conf}	Med_{Lift}	Med_{Conv}	Med_{FC}	$Med_{Netconf}$	$Med_{Yule'sQ}$	Med_{Amp}	*%Reg*
EARMGA	93	0,33	**1**	1,16	∞	0,09	0,01	0,04	**2,03**	**99,2**
GAR	52,1	0,56	0,88	1,45	∞	0,42	0,33	0,58	2,10	84,81
GENAR	29,46	0,24	0,83	3,38	∞	0,57	0,39	0,66	28,96	65,97
Alatasetal	21,94	0,25	0,61	11,38	∞	0,32	0,16	0,24	5,65	43,78
ARMMGA	1	**0,66**	0,91	1,34	∞	0,46	0,35	0,62	**2,03**	66,36
MODENAR	48,52	0,31	0,80	187,31	∞	0,53	0,27	0,56	16,82	62,17
MOEA_Ghosh	**150,13**	0,39	0,61	133,529	∞	0,31	0,19	0,35	23,43	55,61
QAR-CIP-NSGA-II	147,26	0,13	0,93	**440,69**	∞	**0,91**	**0,78**	**0,92**	2,9	92,66

obtenidos por cada método en cada BD se pueden encontrar en la Tablas C.2, C.3, C.4 y C.5 de la sección C.1 del apéndice C. A partir del análisis de los resultados presentados en estas tablas podemos presentar las siguientes conclusiones:

- Los valores obtenidos por nuestra propuesta en las medidas lift, FC, netconf y yule'sQ son mejores que los valores obtenidos por los algoritmos analizados en la mayoría de las BDs, con valores próximos al mejor valor posible que estas medidas pueden alcanzar, lo que nos permite obtener un conjunto interesante de reglas de asociación.
- El método propuesto presenta un buen equilibrio entre todas las medidas que han sido analizadas. Por otra parte, los conjuntos de reglas obtenidos tienen pocos atributos, lo que permite una mejor comprensión del usuario, además obtenemos un alto cubrimiento para la mayoría de las BDs.

Hemos utilizado tests no paramétricos para comparaciones múltiples con el objetivo de encontrar el mejor enfoque entre los algoritmos analizados. Al igual que en la subsección 2.2.2, se aplicaron los tests estadísticos a los resultados medios obtenidos por los algoritmos analizados para las medidas de interés lift, FC, netconf y yule'sQ (son las mismas utilizadas en la sección anterior). En este caso hemos empleado el test de Friedman [Fri37] y el test de Iman y Davenport [ID80] para comprobar si los resultados obtenidos por los algoritmos presentan diferencias significativas. Si existen diferencias significativas, aplicamos el test de Holm [Hol79] y el test de Finner [Fin93] para comparar el algoritmo de mejor ranking con el resto de algoritmos. Utilizamos $\alpha = 0.05$ como nivel de confianza en todos los casos. Una descripción de estos tests y el software para su uso se puede encontrar en: http://sci2s.ugr.es/sicidm/.

La Tabla 2.7 muestra los resultados estadísticos de Friedman e Iman-Davenport, y los relaciona con sus correspondientes valores críticos para cada distribución utilizando el nivel de confianza $\alpha = 0.05$. El p-valor obtenido también se muestran para cada test. Teniendo en cuenta que los resultados estadísticos de Friedman e Iman-Davenport son claramente superiores a sus valores críticos asociados, podemos afirmar que existen diferencias significativas entre los resultados observados con un nivel de confianza $\alpha \leq 0.05$. La Tabla 2.8 muestra los rankings (que se calculan mediante el uso del test de Friedman) de los diferentes métodos que se consideran en este estudio. Como podemos observar, nuestra propuesta obtiene el mejor ranking para todas las medidas analizadas.

Tabla 2.7: Resultados del test de Friedman e Iman-Davenport ($\alpha = 0.05$) en la comparación entre los algoritmos evolutivos y QAR-CIP-NSGA-II

Test de Friedman				
	FC	**Netconf**	**Yule'sQ**	**Lift**
Valor Crítico	14.0671	14.0671	14.0671	14.0671
Estadística (X_F^2)	**78.96**	**92.92**	**92.83**	**109.35**
p **valor**	<0.0001	<0.0001	<0.0001	<0.0001
Test de Iman-Davenport				
	FC	**Netconf**	**Yule'sQ**	**Lift**
Valor Crítico	2.01	2.01	2.01	
Estadística (F_F)	**19.16**	**26.08**	**26.03**	**37.63**
p **valor**	<0.0001	<0.0001	<0.0001	<0.0001

Tabla 2.8: Ranking promedio de los algoritmos evolutivos en la comparación con QAR-CIP-NSGA-II

FC		**Netconf**		**Yule'sQ**		**Lift**	
Algoritmos	*Ranking*	*Algoritmos*	*Ranking*	*Algoritmos*	*Ranking*	*Algoritmos*	*Ranking*
EARMGA	6.90	EARMGA	7.13	EARMGA	7.25	EARMGA	6.69
Alatasetal	5.44	Alatasetal	6	Alatasetal	6.13	ARMMGA	6.34
MOEA_Ghosh	5.28	MOEA_Ghosh	5.19	MOEA_Ghosh	5.55	GAR	5.55
GAR	4.84	MODENAR	4.23	MODENAR	4.13	Alatasetal	5.51
ARMMGA	4.61	GAR	4.17	ARMMGA	4	GENAR	4.5
GENAR	3.76	ARMMGA	4.11	GAR	3.96	MODENAR	3.32
MODENAR	3.75	GENAR	4	GENAR	3.26	MOEA_Ghosh	2.82
QAR-CIP-NSGA-II	**1.38**	QAR-CIP-NSGA-II	**1.15**	QAR-CIP-NSGA-II	**1.69**	QAR-CIP-NSGA-II	**1.23**

La Tabla 2.9 presenta los resultados del test de Holm y el test de Finner para comparar el algoritmo de mejor ranking (QAR-CIP-NSGA-II) con los algoritmos restantes. En esta tabla, los métodos se ordenan con respecto al z-valor obtenido. El test de Holm y el test de Finner rechazan la hipótesis de igualdad con el resto de los métodos ($p < \alpha/i$) en todas las medidas. Por lo tanto, podemos concluir que QAR-CIP-NSGA-II es el método que presenta mejor rendimiento en comparación con los enfoques restantes utilizados en este estudio.

Tabla 2.9: Resultados de los tests de Holm y Finner ($\alpha = 0.05$) en la comparación con los algoritmos evolutivos

i	Algoritmos	z	p	Holm	Finner	**Hipótesis**
			FC			
7	EARMGA	8.124088	0	0.007143	0.007301	**Rechazada**
6	Alatasetal	5.972761	0	0.008333	0.014548	**Rechazada**
5	MOEA-Ghosh	5.746306	0	0.01	0.021743	**Rechazada**
4	GAR	5.095247	0	0.0125	0.028885	**Rechazada**
3	ARMMGA	4.755564	0.000002	0.016667	0.035975	**Rechazada**
2	GENAR	3.510059	0.000448	0.025	0.043013	**Rechazada**
1	MODENAR	3.481752	0.000498	0.05	0.05	**Rechazada**
			Netconf			
7	EARMGA	8.803454	0	0.007143	0.007301	**Rechazada**
6	Alatasetal	7.133345	0	0.008333	0.014548	**Rechazada**
5	MOEA-Ghosh	5.944454	0	0.01	0.021743	**Rechazada**
4	MODENAR	4.529108	0.000006	0.0125	0.028885	**Rechazada**
3	GAR	4.444187	0.000009	0.016667	0.035975	**Rechazada**
2	ARMMGA	4.359267	0.000013	0.025	0.043013	**Rechazada**
1	GENAR	4.189425	0.000028	0.05	0.05	**Rechazada**
			Yule'sQ			
7	EARMGA	8.180702	0	0.007143	0.007301	**Rechazada**
6	Alatasetal	6.5389	0	0.008333	0.014548	**Rechazada**
5	MOEA-Ghosh	5.689692	0	0.01	0.021743	**Rechazada**
4	MODENAR	3.59498	0.000324	0.0125	0.028885	**Rechazada**
3	ARMMGA	3.396831	0.000682	0.016667	0.035975	**Rechazada**
2	GAR	3.340217	0.000837	0.025	0.043013	**Rechazada**
1	GENAR	2.321168	0.020278	0.05	0.05	**Rechazada**
			Lift			
7	EARMGA	8.039167	0	0.007143	0.007301	**Rechazada**
6	ARMMGA	7.529642	0	0.008333	0.014548	**Rechazada**
5	GAR	6.369058	0	0.01	0.021743	**Rechazada**
4	Alatasetal	6.312444	0	0.0125	0.028885	**Rechazada**
3	GENAR	4.812177	0.000001	0.016667	0.035975	**Rechazada**
2	MODENAR	3.085455	0.002032	0.025	0.043013	**Rechazada**
1	MOEA-Ghosh	2.349475	0.0188	0.05	0.05	**Rechazada**

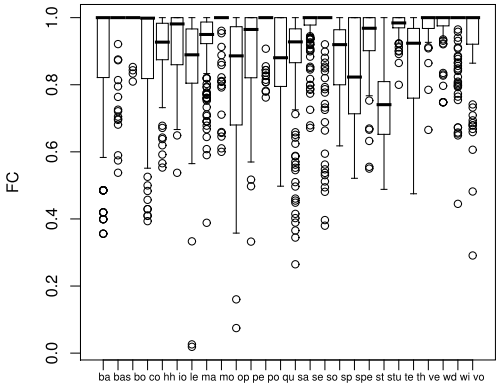

Figura 2.7: Boxplot de la medida FC para QAR-CIP-NSGA-II en todas las BDs

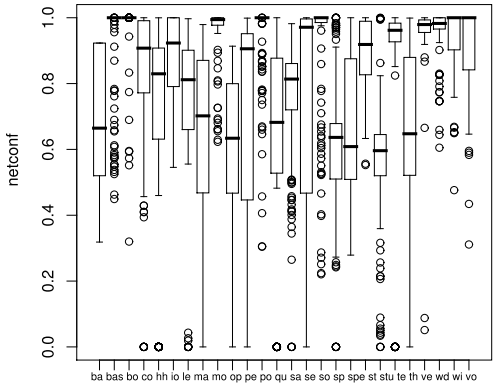

Figura 2.8: Boxplot de la medida netconf para QAR-CIP-NSGA-II en todas las BDs

Las Figuras 2.7 y 2.8 representan boxplots que muestran los valores correspondientes a las medidas FC y netconf, respectivamente, para las reglas obtenidas en una de las 5 ejecuciones realizadas por nuestra propuesta para todas las BDs. Observe que todas las reglas tienen dependencias positivas y más del 75 % de las reglas obtenidas tienen un valor superior a 0.65 para el FC y 0.5 para netconf. Por otra parte, en 4 de las 26 BDs casi el 100 % de las reglas obtenidas por nuestra propuesta obtienen los valores máximos de estas medidas.

Las Figuras 2.9 y 2.10 representan boxplots que muestran los valores de la medida FC y netconf, respectivamente, de las reglas obtenidas en una de las 5 ejecuciones realizadas por todos los algoritmos utilizados en este análisis en la BD stock, seleccionada al azar. Podemos ver que QAR-CIP-NSGA-II presenta los mejores valores de FC y netconf en comparación con el resto de los algoritmos analizados, obteniendo la mayoría de las reglas valores cercanos al mejor valor posible para estas medidas. Observe que MODENAR obtiene algunas reglas con dependencia negativa (FC<0) y que MOEA_Ghosh y varios de los algoritmos mono-objetivo extraen reglas con valores cercanos a 0.

2.2.4. Comparación con algoritmos clásicos para extraer reglas de asociación

En esta sección se analiza el rendimiento de nuestro algoritmo con los algoritmos clásicos Apriori [Bor03, SA96] y Eclat [Zak00]. Como hemos comentado

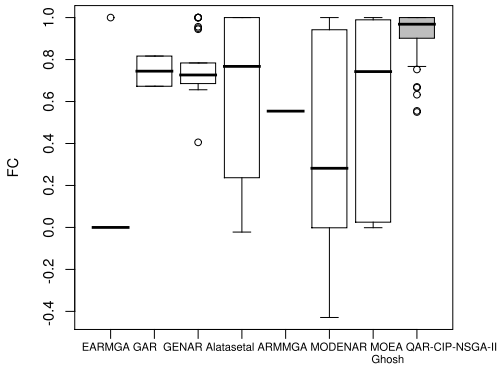

Figura 2.9: Boxplot de la medida FC para los otros algoritmos evolutivos y QAR-CIP-NSGA-II para la BD stock

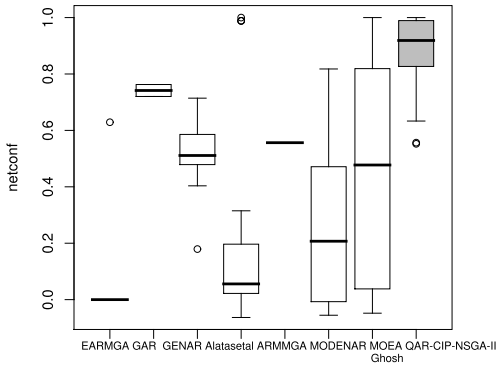

Figura 2.10: Boxplot de la medida netconf para los otros algoritmos evolutivos y QAR-CIP-NSGA-II para la BD stock

anteriormente, un método comúnmente utilizado por los algoritmos clásicos para descubrir RACs es dividir el dominio de las variables cuantitativas en intervalos y considerar cada intervalo como un valor categórico para extraer las reglas de asociación. En este estudio hemos utilizado un particionamiento en amplitud en cada atributo cuantitativo [LHTD02], el cual es uno de los algoritmos de discretización habitual utilizado cuando no tenemos información adicional para utilizar algoritmos basados en la teoría de la información [Lee07, TLY08] u otros conceptos [JBVW06].

La Tabla 2.10 muestra los resultados medios obtenidos por los métodos en todas las BD (la cabecera de esta tabla ha sido introducida en la sección 2.2.2). Los resultados obtenidos por cada método en cada BD se pueden encontrar en la Tabla C.6 de la sección C.1 del apéndice C. Observe que sólo mostramos los resultados de 15 BDs para los métodos Apriori y Eclat debido a los problemas de escalabilidad que ellos presentan por lo que no pueden ejecutarse en todas las BDs. Para analizar los resultados obtenidos para las medidas de calidad hemos utilizado el test de Wilcoxon [She03, Wil45], considerando las medidas FC, netconf, yule'sQ y lift. La Tabla 2.11 muestra los resultados de este test.

Como podemos ver en la Tabla 2.10 los algoritmos clásicos extraen grandes conjuntos de reglas de asociación, obteniendo un buen cubrimiento de las BDs. Por el contrario, nuestra propuesta nos permite obtener un conjunto reducido de reglas con un buen cubrimiento en todas las BDs y un buen equilibrio entre todas

Tabla 2.10: Resultados del valor medio de las medidas para todas las BDs en la comparación entre NIGAR y los métodos clásicos

Algoritmos	*#R*	Med_{Sop}	Med_{Conf}	Med_{Lift}	Med_{Conv}	Med_{FC}	$Med_{Netconf}$	$Med_{Yule'sQ}$	Med_{Amp}	*%Ejem*
Apriori	**8345221,46**	**0,16**	**0,93**	4,09	∞	0,81	0,64	0,84	5,13	90,43
Eclat	**8345221,46**	**0,16**	**0,93**	4,09	∞	0,81	0,64	0,84	5,13	90,43
QAR-CIP-NSGA-II	147,26	0,13	**0,93**	**440,69**	∞	**0,91**	**0,78**	**0,92**	**2,9**	**92,66**

Tabla 2.11: Resultados del test de Wilcoxon ($\alpha = 0.05$) en la comparación entre los algoritmos clásicos de extracción de reglas de asociación y QAR-CIP-NSGA-II

Comparación	R^+	R^-	**Hipótesis**	p**-valor**
		FC		
QAR-CIP-NSGA-II vs. Apriori	115,5	4.5	**Rechazada**	0.0013
QAR-CIP-NSGA-II vs. Eclat	115,5	4.5	**Rechazada**	0.0013
		Netconf		
QAR-CIP-NSGA-II vs. Apriori	106.0	14.0	**Rechazada**	0.006
QAR-CIP-NSGA-II vs. Eclat	106.0	14.0	**Rechazada**	0.006
		Yule'sQ		
QAR-CIP-NSGA-II vs. Apriori	61.0	44.0	**No Rechazada**	≥ 0.2
QAR-CIP-NSGA-II vs. Eclat	61.0	44.0	**No Rechazada**	≥ 0.2
		Lift		
QAR-CIP-NSGA-II vs. Apriori	120.0	0.0	**Rechazada**	<0.0001
QAR-CIP-NSGA-II vs. Eclat	120.0	0.0	**Rechazada**	<0.0001

las medidas que han sido analizadas. Si nos centramos en los resultados obtenidos por el test de Wilcoxon podemos observar como la hipótesis de igualdad ha sido rechazada (p-valor $\leq \alpha$) en todas las medidas excepto para yules'Q, en donde no se consiguen diferencias significativas, pero se obtiene un mayor número de rankings positivos para las 15 BDs en las que los clásicos pudieron ejecutarse.

La Figura 2.11 representa un boxplot que muestra los valores de la medida FC y netconf para las reglas obtenidas por los algoritmos clásicos y una de las 5 ejecuciones realizadas por nuestra propuesta para la BD stock. Podemos ver cómo ninguna de las reglas obtenidas por los algoritmos clásicos logra el mejor valor para estas medidas (promedio 0,84), mientras QAR-CIP-NSGA-II obtiene en la mayoría de las reglas, el mejor valor posible que pueden alcanzar estas medidas.

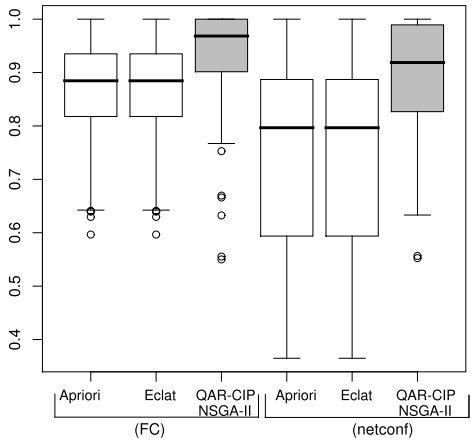

Figura 2.11: Boxplot de la medida FC y netconf para los algoritmos clásicos y QAR-CIP-NSGA-II para la BD stock

2.2.5. Análisis de escalabilidad

Varios experimentos han sido realizados para analizar la escalabilidad de los algoritmos en la BD House_16H. Todos los experimentos se realizaron en un procesador Intel Core i7, 2,80 GHz CPU con 12 Gb de memoria y fueron ejecutados en Linux. El tiempo de ejecución promedio empleado por los algoritmos analizados cuando aumentan el número de atributos y ejemplos se muestran en la Tabla 2.12 y Tabla 2.13, respectivamente.

La Figura 2.12 muestra la relación entre el tiempo de ejecución y el número de atributos para nuestra propuesta y los algoritmos mono-objetivo y multi-objetivo estudiados. Podemos ver cómo la mayoría de los algoritmos tienden a escalar linealmente cuando el número de atributos de la BD aumenta. La Figura 2.13 muestra la relación entre el tiempo de ejecución y el número de ejemplos para los mismos algoritmos. Al igual que en el caso anterior, el tiempo de ejecución tiende a escalar linealmente cuando el número de ejemplos de la BD aumenta. Observe que los tiempos de ejecución de GAR son más altos que el resto de los algoritmos evolutivos porque necesita un proceso adicional para extraer las reglas de asociación.

Tabla 2.12: El tiempo de ejecución (segundos) empleado por todos los algoritmos comparados y QAR-CIP-NSGA-II cuando el número de atributos aumenta en la BD House_16H

	Número de Atributos				
Algoritmos	4	8	12	16	17
EARMGA	78	61	78	75	65
GAR	619	1528	1991	2083	2014
GENAR	24	38	**47**	57	**59**
Alatasetal	50	77	52	72	154
ARMMGA	104	93	95	100	97
MODENAR	60	81	97	95	93
MOEA_Ghosh	23	36	50	**46**	73
Apriori	**3**	**5**	233	5192	11268
Eclat	**3**	**5**	251	5812	12467
QAR-CIP-NSGA-II	34	46	59	66	70

Tabla 2.13: El tiempo de ejecución (segundos) empleado por todos los algoritmos comparados y QAR-CIP-NSGA-II cuando el número de ejemplos aumenta en la BD House_16H

	Número de Ejemplos				
Algoritmos	20 %	40 %	60 %	80 %	100 %
EARMGA	15	31	39	50	65
GAR	467	898	1260	1595	2014
GENAR	**12**	**23**	**36**	47	**59**
Alatasetal	16	24	93	104	154
ARMMGA	21	40	60	77	97
MODENAR	13	41	37	115	93
MOEA_Ghosh	13	24	37	**44**	73
Apriori	2689	5180	10050	9004	11268
Eclat	3076	8700	11300	10604	12467
QAR-CIP-NSGA-II	**12**	27	37	50	71

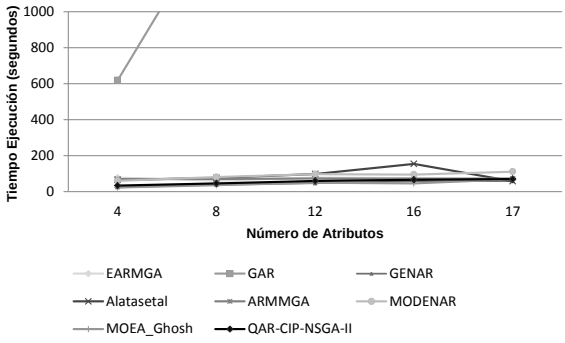

Figura 2.12: Relación entre el tiempo de ejecución y el número de atributos en la BD House_16H para los algoritmos evolutivos y QAR-CIP-NSGA-II

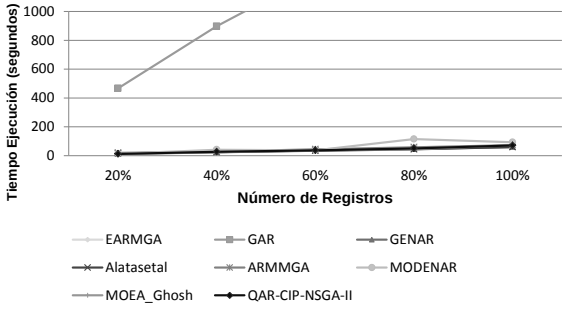

Figura 2.13: Relación entre el tiempo de ejecución y el número de ejemplos en la BD House_16H para los algoritmos evolutivos y QAR-CIP-NSGA-II

La Figura 2.14 muestra la relación entre el tiempo de ejecución y el número de atributos de nuestra propuesta y los algoritmos clásicos. Podemos ver como el tiempo de ejecución de los algoritmos clásicos aumenta exponencialmente cuando el número de atributos es superior a 10. La Figura 2.15 muestra la relación entre el tiempo de ejecución y el número de ejemplos para estos algoritmos. Se puede ver fácilmente cómo el tiempo de ejecución de los algoritmos clásicos es muy alto cuando el número de ejemplos de la BD aumenta.

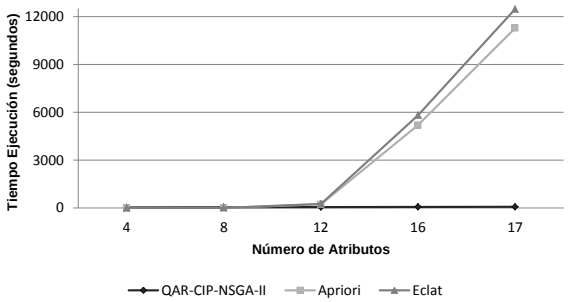

Figura 2.14: Relación entre el tiempo de ejecución y el número de atributos en la BD House_16H para los algoritmos clásicos y QAR-CIP-NSGA-II

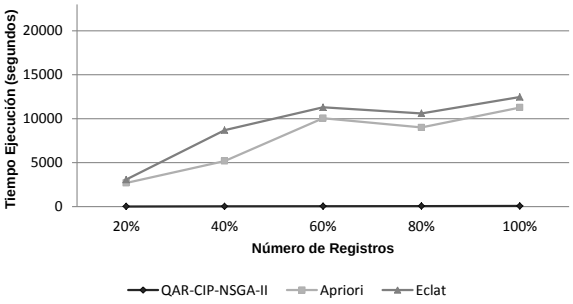

Figura 2.15: Relación entre el tiempo de ejecución y el número de ejemplos en la BD House_16H para los algoritmos clásicos y QAR-CIP-NSGA-II

2.3. Sumario

En este capítulo hemos propuesto QAR-CIP, un nuevo modelo evolutivo multi-objetivo para extraer un conjunto de RACs de muy buena calidad, con un buen equilibrio entre las diferentes medidas de interés, maximizando tres objetivos: interés, comprensión y rendimiento. Para lograr esto, hemos propuesto el algoritmo QAR-CIP-NSGA-II, el cual extiende el AEMO NSGA-II para realizar un aprendizaje evolutivo de los intervalos de los atributos y una selección de condiciones para cada regla. Esta propuesta introduce dos nuevos componentes al

proceso evolutivo: un proceso de reinicialización y una PE para almacenar todas las reglas no dominadas encontradas y promover diversidad en la población. Del estudio realizado podemos sacar las siguientes conclusiones:

- El proceso de reinicialización y la PE mejoran el porciento de ejemplos cubiertos por las reglas respecto al total de ejemplos de la BD y nos permiten obtener un número mayor de reglas que el enfoque clásico, ya que el número de reglas obtenidas no está limitado por el tamaño de la población actual.
- Además, al comparar los resultados obtenidos con otros algoritmos mono-objetivo y multi-objetivo y los algoritmos clásicos podemos ver cómo QAR-CIP-NSGA-II nos permite extraer un conjunto de reglas con un buen equilibrio entre los diferentes objetivos, obteniendo reglas de asociación con mejores valores para las medidas de interés, un alto cubrimiento de la BD y pocos atributos, proporcionando al usuario reglas de alta calidad.
- Por último, el enfoque propuesto emplea una cantidad razonable de tiempo en todas las BDs.

Capítulo 3

MOPNAR: Algoritmo Evolutivo Multi-Objetivo para Extraer Reglas de Asociación Cuantitativas Positivas y Negativas

Como comentamos en el capítulo 1, los métodos de extracción de reglas de asociación han sido normalmente enfocados para generar reglas de asociación positivas sin tener en cuenta las reglas de asociación negativas, las cuales también son interesantes para el usuario al indicar la ausencia de un conjunto de ítems ante la presencia de otros. Además, las reglas negativas permiten representar conocimiento de la BD para el que se necesitaría varias reglas positivas para representarlo, permitiendo reducir el número de reglas suministrado al usuario pero representando el mismo conocimiento. Debido a ellos algunos trabajos han sido propuestos para extraer reglas de asociación positivas y negativas a partir de BD con datos cuantitativos [AA06, BMS97, TRD12, WZZ04].

Como hemos visto en el capítulo anterior, los AEMOs proporcionan una forma interesante de extraer reglas de asociación al permitirnos optimizar conjuntamen-

te varios objetivos durante el proceso de extracción, proporcionando al usuario un conjunto de reglas del frente Pareto, donde cada regla presenta un grado diferente de equilibrio entre los distintos objetivos optimizados.

Recientemente ha crecido el interés por los AEMOs basados en descomposición (MOEA/D [ZL07] y MOEA/D-DE [LZ09]), los cuales explícitamente descomponen el problema de optimización multi-objetivo en N subproblemas de optimización escalares y los optimizan de manera simultánea. Estos enfoques han mostrado algunas ventajas sobre otros AEMOs, presentando menor complejidad computacional y un mejor funcionamiento en los problemas continuos de 3-objetivos. Destacar que MOEA/D ganó la competición del CEC2009. Estas razones han suscitado un creciente interés por estos enfoques dentro de la comunidad de los AEMOs.

En este capítulo presentamos MOPNAR, un nuevo AEMO basado en el modelo MOEA/D-DE [LZ09] que nos permite extraer un conjunto reducido de RACPNs con un buen equilibrio entre el número de reglas, el soporte y el cubrimiento de la BD. Este nuevo método realiza un aprendizaje evolutivo de los intervalos de los atributos y una selección de las condiciones para cada regla, maximizando tres objetivos: rendimiento, interés y comprensibilidad. Además, al igual que en el método descrito en el capítulo 2, hemos introducido un proceso de reinicialización y una PE al modelo evolutivo, con el fin de promover diversidad en la población, almacenar todas las reglas no dominadas encontradas y mejorar el cubrimiento de la BD.

El capítulo se organiza en las siguientes secciones. La sección 3.1 detalla el algoritmo evolutivo propuesto para obtener reglas de asociación positivas y negativas. En la sección 3.2 se muestran los resultados obtenidos sobre las 26 BDs reales utilizadas en el estudio. Por último, se incluye un breve resumen del capítulo en la sección 3.3.

3.1. Algoritmo Evolutivo Multi-Objetivo para extraer Reglas de Asociación Cuantitativas Positivas y Negativas: MOPNAR

En esta sección se describe nuestra propuesta, MOPNAR, para obtener un conjunto de reglas positivas y negativas de buena calidad con un buen equilibrio entre el número de reglas y el cubrimiento de la BD. Esta propuesta extiende

el algoritmo MOEA/D-DE [LZ09] para realizar un aprendizaje evolutivo de las reglas, introduciendo a su modelo evolutivo: un proceso de reinicialización y una PE. A continuación, explicamos en detalle todas sus características.

3.1.1. Modelo Evolutivo Multi-Objetivo MOEA/D-DE

Nuestra propuesta extiende el modelo evolutivo basado en descomposición MOEA/D-DE [LZ09] para extraer reglas de asociación positivas y negativas. Esta propuesta descompone el problema de optimización multi-objetivo en N subproblemas de optimización escalares con el propósito de cada uno de estos subproblemas optimice una agregación distinta de todos los objetivos. Para almacenar todas las reglas no dominadas generadas hemos introducido una PE al modelo evolutivo (ver subsección 2.1.1 del capítulo 2). La PE se actualiza en cada generación con los hijos generados para cada solución, manteniendo las reglas no dominadas de la población. Las reglas redundantes se eliminarán de la PE para evitar solapamiento entre las reglas.

Además, para no caer en óptimos locales y provocar diversidad en la población, el proceso de reinicialización será aplicado cuando el número de nuevas soluciones en la población es menor que $\alpha\,\%$ ($\alpha\,\%$ definido por el usuario, normalmente al $5\,\%$) (ver subsección 2.1.1 del capítulo 2). En este caso la población se reinicia basándose en los registros que no hayan sido cubiertos por las reglas de la PE y se actualiza la PE con la nueva población (ver subsección 3.1.2). Este proceso nos permite realizar una buena exploración del espacio de búsqueda y mejorar el cubrimiento de las BDs.

Con las modificaciones anteriores, el modelo evolutivo sería el siguiente. Este modelo primero genera un vector de pesos para cada subproblema, que son utilizados para calcular el valor del enfoque de descomposición de cada subproblema. Luego se selecciona un conjunto de vecinos para cada vector de pesos, conteniendo los T vectores de pesos más cercanos. Después el algoritmo genera una población inicial, inicializa el punto de referencia para cada objetivo con los mejores valores encontrados hasta el momento e inicializa la PE con las reglas no dominadas de la población inicial. A continuación se generan dos hijos aplicando los operadores de cruce, mutación y reparación. Estos se aplican sobre una solución de la población y sobre otra solución seleccionada aleatoriamente con una probabilidad δ de su conjunto de vecinos o de la población (δ es definida por el usuario). Estos hijos se utilizan para actualizar los puntos de referencias y sustituir algunas soluciones de la población actual que tengan los peores valores del enfoque de descomposición.

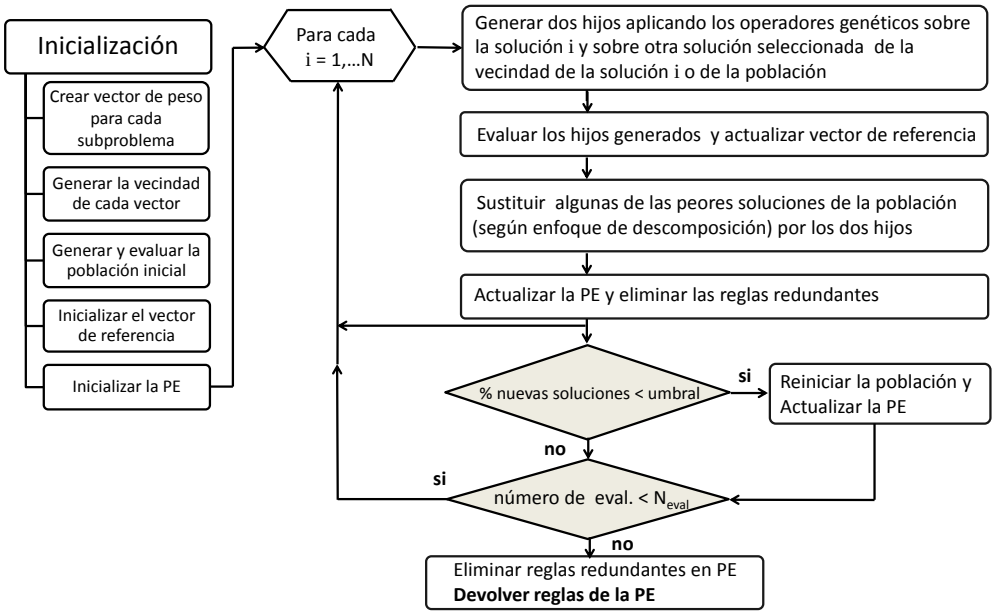

Figura 3.1: Organigrama del método MOPNAR

Téngase en cuenta que el máximo número de soluciones que se pueden sustituir por una solución hija es limitado y debe ser mucho menor que T. Estos pasos se repiten para cada solución de la población, se actualiza la PE y si es necesario se aplica el proceso de reinicialización. Todo este proceso se repite hasta que se cumpla la condición de parada (ver [LZ09] para más información).

En [LZ09] los autores presentaron tres enfoques de descomposición donde recomendaron el propuesto por Tchebycheff [Mie99], el cual es el utilizado en este trabajo.

La Figura 3.1 muestra un esquema del modelo evolutivo.

3.1.2. Esquema de Codificación y Población Inicial

El esquema de codificación utilizado en esta propuesta extiende el descrito en la subsección 2.1.2 del capítulo 2, agregando una cuarta parte a la codificación para representar reglas de asociación positivas y negativas. Así, cada gen constará de 4 partes: *ac* indica si un gen es considerado en la regla, si es parte del

antecedente, del consecuente o no está involucrado en la regla; *pn* indica si el intervalo es positivo o negativo, cuando esta parte es '0' el intervalo es negativo y cuando es '1' el intervalo es positivo; *li* y *ls* representan el límite inferior y superior del intervalo respectivamente. Por tanto un cromosoma C_T se codifica de la siguiente manera:

$$C_T = Gen_1 Gen_2 \dots Gen_n, \;\; i = 1, \dots, n$$
$$Gen_i = (ac_i, pn_i, li_i, ls_i)$$

Por ejemplo, consideremos una BD simple con cuatro atributos X_1, X_2, X_3 y X_4. Supongamos que seleccionamos al azar los atributos X_1 y X_3 para el antecedente y X_4 para el consecuente de la regla. Sobre la base de esta definición, la [illegible] $\in [5, 25]$ y $X_3 \in \neg[$[illegible]

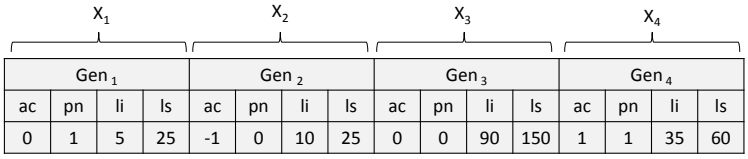

Figura 3.2: Ejemplo de un cromosoma codificado por MOPNAR

Para evitar que los intervalos crezcan hasta cubrir el dominio completo de los atributos, también hemos utilizado *Amplitud* (vea ecuación 2.1 del capítulo 2). De esta manera, ningún intervalo positivo podrá tener un tamaño mayor que *Amplitud*, por otro lado, ningún intervalo negativo podrá tener un tamaño menor que *Amplitud*.

La población inicial estará compuesta por un conjunto de reglas que contienen un solo atributo en el consecuente y presentan un buen cubrimiento de la BD, utilizando para ello el mismo proceso descrito en la subsección 2.1.2 del capítulo 2. En este proceso, primero se selecciona aleatoriamente los atributos que formarán parte del antecedente y del consecuente de la regla. Después se selecciona si el intervalo será positivo o negativo. Luego, se selecciona aleatoriamente un ejemplo de la BD para crear un intervalo de manera que el valor del ejemplo seleccionado quede en el centro del intervalo, con un tamaño para el intervalo igual al 50 % de *Amplitud* de su atributo. Además, si alguno de los límites del intervalo supera el límite del dominio de su atributo, entonces se reemplaza el límite del intervalo

por el límite del dominio del atributo. Finalmente, se marcan los ejemplos de la BD que cubre la regla. Este proceso se repite hasta que se complete la población inicial utilizando los ejemplos de la BD que no han sido marcados. Si se han marcado todos los ejemplos de la BD y la población inicial no se ha completado, se vuelven a desmarcar y el proceso se repite hasta que esta sea completada. Por último, se inicializa la PE con la reglas no dominadas de la población inicial.

3.1.3. Operadores

Los operadores de cruce, mutación y reparación utilizados en esta propuesta se basan en los descritos en la subsección 2.1.4 del capítulo 2. Como hemos comentado el operador de cruce genera dos hijos intercambiando aleatoriamente los genes de dos padres seleccionados.

El operador de mutación modifica aleatoriamente los componentes de un gen seleccionado al azar. Primero, modifica el valor de *ac* aleatoriamente, dentro del conjunto {-1,0,1} y luego el valor de *pn* dentro del conjunto {0,1}. Finalmente se selecciona al azar uno de los límites del intervalo modificando su valor de manera aleatoria para aumentar o disminuir el tamaño del intervalo. Para disminuir el intervalo se selecciona un nuevo valor aleatoriamente entre el valor actual del límite y el valor del otro límite del intervalo. Por otro lado, si se va a aumentar el intervalo el valor será seleccionado entre el valor actual del límite y el límite del dominio del atributo. Destacar que si la longitud del intervalo supera el valor de *Amplitud* del atributo entonces será reajustado el nuevo valor del intervalo para que no se supere.

El operador de reparación evita reglas que tengan más de un atributo en el consecuente o que no tengan ningún atributo en el antecedente o consecuente. En el proceso de obtener reglas más simples decrementa el tamaño de los intervalos mientras que se cubran los mismos ejemplos cubiertos por los intervalos originales. Destacar que en el caso de que el intervalo sea negado, el intervalo será incrementado (reduciendo el dominio que cubre) mientras los ejemplos cubiertos sean los mismos. La Figura 3.3 muestra un ejemplo de este proceso para ajustar los límites de un intervalo negativo $X_1 \in \neg[7, 20]$, considerando una BD simple de tres atributos y 6 ejemplos, donde la regla a la que pertenece dicho intervalo cubre los ejemplos de la BD que están marcados en gris en la tabla de la figura.

Observe que hemos utilizado operadores genéticos comunes que funcionan bien

para extraer reglas de asociación en vez de los operadores genéticos del modelo MOEA/D-DE diseñados para problemas de optimización multi-objetivo.

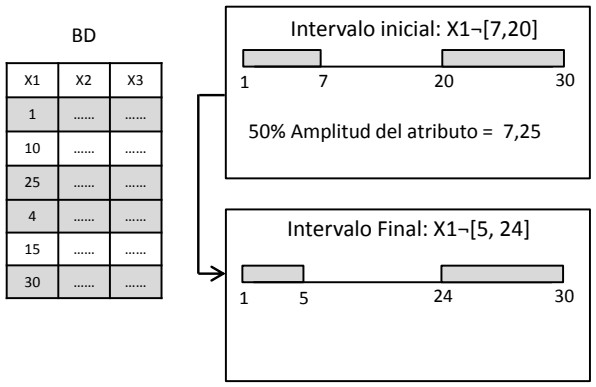

Figura 3.3: Ejemplo del proceso de ajuste de los límites de un intervalo negativo

3.1.4. Objetivos

Por último, nuestra propuesta maximiza tres objetivos: rendimiento, interés y comprensibilidad descritos en la subsección 2.1.3 del capítulo 2. Como se ha dicho, el rendimiento es definido como el producto entre el soporte y el FC, lo que nos permite obtener un conjunto de reglas con un buen equilibrio entre reglas específicas y generales. Destacar que esta propuesta solo genera reglas con dependencia fuerte [BBSV02] entre los ítems porque representan dependencias positivas entre ellos y evitan las limitaciones que presenta el soporte (ver sección 1.2 del capítulo 1). Además, este método solo genera reglas con dependencia fuerte [BBSV02] entre los ítems porque representan dependencias positivas entre ellos y evitan el problema del soporte (ver sección 1.2.2 del capítulo 1), por lo que solo obtendremos aquellas reglas con FC > 0.

Además, también hemos usado la medida de interés lift (ver sección 1.2.1 del capítulo 1), porque nos permite detectar dependencias negativas, positivas o independencia entre los ítems y reducir el número de empates entre las reglas al poder establecer mayores diferencias entre ellas, debido a que el rango de valores de esta medida no está limitado.

Finalmente, la medida de comprensibilidad nos permite medir cuán fáciles de interpretar son las reglas, teniendo en cuenta el número de atributos que involucran (ver ecuación 2.2 del capítulo 2).

3.1.5. Pasos del algoritmo

De acuerdo con la descripción anterior, el algoritmo propuesto para extraer RACPNs se puede resumir en los siguientes pasos:

Entrada:

- N tamaño de la población
- N_{eval} número de evaluaciones
- m número de objetivos
- P_{mut} probabilidad de mutación
- $\lambda^1, ...\lambda^N$ conjunto de N vectores de pesos
- T número de vectores de pesos en el conjunto de vecinos de cada vector de pesos
- δ probabilidad de que las soluciones padres sean seleccionadas del conjunto de vecinos
- η_r número máximo de soluciones reemplazadas por cada solución hijo
- γ factor de amplitud para cada atributo de la BD
- α umbral de diferencia

Salida: Reglas de la PE

Paso 1: Inicialización:

1.1 Calcular las distancias euclidianas entre cada vector de pesos (para cada $i = 1, ..., N$) y el resto de los vectores con el fin de crear el conjunto de vecinos de cada vector λ^i. La vecindad de un vector son los T vectores de peso más cercanos, donde $B_i = \{i_i, ...i_T\}$ representa la vecindad y $\lambda^{i_1}, ...\lambda^{i_T}$ son los T vectores de peso más cercanos a λ^i.

1.2 Generar la población inicial con N cromosomas.

1.3 Evaluar la población inicial.

1.4 Inicializar el vector de puntos de referencia $z = (z_1, ..., z_m)$ asignando $z_j = max_{1 \leq i \leq N} f_j(x^i)$, donde $j = 1, ..., m$

1.5 Inicializar la PE.

Paso 2: Para cada $i = 1, ..., N$ hacer

2.1 Generar un valor aleatorio *rand* en el intervalo [0‘0, 1‘0]. A partir del valor rand, asignamos aleatoriamente a P una solución del siguiente conjunto de soluciones de la siguiente manera:

$$P = \begin{cases} \text{B(i)} & \text{si } rand < \delta \\ poblacioncompleta & \text{sino} \end{cases}$$

2.2 Asignar $r_1 = i$ y seleccionar aleatoriamente r_2 de P. Las soluciones x^{r_1} y x^{r_2} se cruzarán, generando dos hijos y_1 and y_2. A continuación, los operadores de mutación y reparación se aplican para los dos hijos.

2.3 Evaluar los nuevos individuos. Para cada y_k, $k \in \{1, 2\}$:

2.3.1 Actualizar z: Para cada $j = 1, ...m$, si $z_j < f_j(y_k)$, entonces asignar $z_j = f_j(y_k)$

2.3.2 Actualizar las soluciones y luego hacer lo siguiente:

a) asignar $c = 0$

b) si $c == \eta_r$ o P está vacío ir al Paso 3. De lo contrario, elegir al azar un índice l de P.

c) si $g(y_k|\lambda^l, z) \leq g(x^l|\lambda^l, z)$, entonces $x^l = y_k$ y $c = c + 1$

d) Eliminar l de P y vaya a b)

Paso 3: Actualizar la PE: eliminar de la PE todas las soluciones dominadas por i ($i = 1, ..., N$), y añadir la solución i a la PE si no hay soluciones en la PE que la dominan.

Paso 4: Eliminar las reglas redundantes de la PE.

Paso 5: Si la diferencia entre la población actual y la población anterior es menor que $\alpha\,\%$, entonces se reinicia la población.

Paso 6: Si no se ha alcanzado el número máximo de evaluaciones, ir al Paso 2.

Paso 7: Eliminar las reglas redundantes de la PE.

Paso 8: Devolver las reglas de la PE.

3.2. Estudio Experimental

Para evaluar el comportamiento del método propuesto hemos realizado varios experimentos considerando 26 BDs del mundo real. Para presentar todos los experimentos realizados, esta sección ha sido organizada como sigue:

- En la subsección 3.2.1 se presentan las BDs usadas en estos experimentos y la configuración de los parámetros de los algoritmos que utilizaremos en este estudio.
- En la subsección 3.2.2 se compara nuestro enfoque con el algoritmo Alatasetal [AA06], ya que este algoritmo evolutivo puede extraer reglas positivas y negativas.
- En la subsección 3.2.3 se compara el rendimiento de nuestra propuesta con tres algoritmos genéticos mono-objetivos (EARMGA [YZZ09], GAR [MAR02] y GENAR [MAR01]) y tres AEMOs (ARMMGA [QNMB11], MODENAR [AA08] y MOEA_Ghosh [GN04]) para extraer reglas de asociación.
- En la subsección 3.2.4 se compara nuestro método con dos algoritmos clásicos para extraer reglas de asociación positivas (Apriori [Bor03, SA96] y Eclat [Zak00]) y con otro AEMO clásico (NSGA-II [DAPM02]).
- En la subsección 3.2.5 se analiza la escalabilidad de nuestra propuesta.
- En la subsección 3.2.6 se presenta un estudio de algunas de las reglas obtenidas por nuestra propuesta.

3.2.1. Experimentos

Para realizar estos experimentos hemos utilizado las mismas 26 BDs utilizadas en el capítulo 2 (ver subsección 2.2.1 del capítulo 2), las cuales pueden ser descargadas del repositorio de datos KEEL-dataset [AFFL$^+$11]. Para desarrollar los diferentes experimentos, consideramos los resultados promedio de 5 ejecuciones para cada BD.

En este estudio comparamos nuestra propuesta con un Alatasetal [AA06], un algoritmo genético que permite extraer reglas de asociación positivas y negativas, además nos comparamos con los algoritmos genéticos mono-objetivo EARMGA [YZZ09], GAR [MAR02] y GENAR [MAR01] y los AEMOs ARMMGA

Tabla 3.1: Parámetros considerados por los algoritmos analizados

Algoritmos	*Parámetros*
Apriori	*mínimo de soporte = 0,1, mínimo de confianza = 0,8*
Eclat	*mínimo de soporte = 0,1, mínimo de confianza = 0,8*
Alatasetal	N_{eval}=50000, nCromoInicialAleat=12, r = 3, TamTorneo = 10, P_{sel}= 0,25, P_{cru} = 0,7, P_{mut_min} = 0,05, P_{mut_max} = 0,9, $Peso_{sop}$ = 5, $Peso_{conf}$ = 20, $Peso_{amplRule}$ = 0,05, $Peso_{amplInterv}$ = 0,02, $Peso_{cubrimiento}$ = 0,01
EARMGA	*TamPop = 100,* N_{eval} *= 50000, k = 2,* P_{sel} *= 0,75,* P_{cru} *= 0,7,* P_{mut} *= 0,1,* α *= 0,01*
GENAR	*TamPop = 100,* N_{eval} *= 50000,* P_{sel}*= 0,25,* P_{cru} *= 0,7,* P_{mut} *= 0,1, nReglas = 30, FP = 0,7, AF = 0,2*
GAR	*TamPop = 100, nItems = 100,* N_{eval} *= 50000,* P_{sel}*= 0,25,* P_{cru} *= 0,7,* P_{mut} *= 0,1,* ω *= 0,4,* Ψ*= 0,7,* μ*= 0,5, mínimo de soporte = 0,1, mínimo de confianza = 0,8*
ARMMGA	*TamPop=100,* N_{eval}*=50000, k=2 (3 with HH),* P_{sel}*=0,95,* P_{cru}*=0,85,* P_{mut} *= 0,01, db=0,01*
MODENAR	*TamPop = 100,* N_{eval}*=50000, Umbral= 60, CR = 0,3,* $Peso_{sop}$ *= 0,8,* $Peso_{conf}$ *= 0,2,* $Peso_{comp}$ *= 0,1,* $Peso_{amplInterv}$ *= 0,4*
MOEA_Ghosh	*TamPop = 100,* N_{eval}*=50000, PuntoCruce=2,* P_{cru}*=0,8,* P_{mut}*= 0,02*
NSGA-II	*TamPop = 100,* N_{eval}*=50000,* γ*=2,* P_{mut}*= 0.1*
MOPNAR	N_{eval}*=50000, H=13, m=3, TamPop=*N^{m-1}_{H+m-1}*, T=10,* δ*=0.9,* η_r*=2,* γ*=2,* P_{mut}*= 0.1,* α *= 5 %*

[QNMB11], MODENAR [AA08] y MOEA_Ghosh [GN04] (ver subsección 1.4.1 y 1.4.2 del capítulo 1, respectivamente). Además, comparamos nuestra propuesta con los algoritmos clásicos Apriori [Bor03, SA96] y Eclat [Zak00] (ver subsección 1.3 del capítulo 1) y con el AEMO clásico NSGA-II [DAPM02] (descrito en el capítulo anterior). La configuración de los parámetros que hemos utilizado para estos métodos se muestra en la subsección 2.2.1 del capítulo 2.

La Tabla 3.1 muestra los parámetros de los métodos analizados. Con estos valores para nuestra propuesta, hemos tratado de facilitar las comparaciones, seleccionando parámetros estándar comunes que funcionan bien en la mayoría de los casos y para el resto de los algoritmos los hemos seleccionado de acuerdo a las recomendaciones de los autores de cada propuesta.

3.2.2. Comparación con el algoritmo propuesto por Alatas para extraer reglas de asociación positivas y negativas

En esta sección se estudia el rendimiento de nuestra propuesta en comparación con el enfoque evolutivo para extraer RACPNs propuesto por Alatas [AA06]. La Tabla 3.2 muestra los resultados medios obtenidos por ambos métodos en todas

Tabla 3.2: Resultados del valor medio de las medidas para todas las BDs en la comparación entre MOPNAR y Alatasetal

Algoritmos	$\#R$	Med_{Sop}	Med_{Conf}	Med_{Lift}	Med_{Conv}	Med_{FC}	$Med_{Netconf}$	$Med_{Yule'sQ}$	Med_{Amp}	$\%Ejem$
Alatasetal	21,94	0,25	0,61	11,38	∞	0,32	0,16	0,24	5,65	43,78
MOPNAR	**70,98**	**0,28**	**0,91**	**12,39**	∞	**0,87**	**0,7**	**0,96**	**2,92**	**99,49**

las BD, donde $\#R$ representa el número medio de reglas, Med_{Sop}, Med_{Conf}, Med_{Lift}, Med_{Conv}, Med_{FC}, $Med_{Netconf}$, Med_{YulesQ} representan los valores medios de soporte, confianza, lift, conviction, FC, netconf y yule'sQ, respectivamente, Av_{Amp} el número medio de atributos involucrados en las reglas y $\%Ejem$ es el tanto por ciento de ejemplos de la BD cubiertos por las reglas generadas. Además, en la Tabla C.7 de la sección C.2 del apéndice C se pueden encontrar los resultados obtenidos por cada método en cada BD. A partir de analizar los resultados presentados en estas tablas podemos obtener las siguientes conclusiones:

- Las reglas obtenidas por nuestra propuesta presentan mejoras en casi todas las medidas de interés sobre las reglas obtenidas por Alatasetal en todas las BDs. Alatasetal extrae reglas con buen promedio de soporte y confianza, pero algunas de ellas presentan ítems con soporte alto en el consecuente o dependencias negativas, obteniendo bajos valores para el resto de las medidas.
- Nuestra propuesta obtiene un conjunto reducido de RACPNs con pocos atributos y sin solapamiento entre reglas (menos de 100 en todos las BDs), donde cada regla nos proporciona un conocimiento interesante de la BD. Además, el cubrimiento de las BDs es muy alto (cercano al 100 % en todas las BDs), aportando conocimiento e información sobre toda la BD. Alatasetal obtiene un menor número de reglas que nuestra propuesta pero con valores más bajos de cubrimiento para casi todas las BDs.
- MOPNAR presenta un conjunto reducido de RACPNs interesantes, obteniendo un buen equilibrio entre el número de reglas, el soporte y el cubrimiento.

Para analizar los resultados obtenidos por los dos algoritmos para las medidas de interés FC, netconf, yule'sQ y lift (son las mismas utilizadas en la sección 2.2.2 del capítulo 2) hemos utilizado el test de Wilcoxon [She03, Wil45]. Los resultados del test de Wilcoxon para las cuatro medidas se muestran en la Tabla 3.3, donde

Tabla 3.3: Resultados del test de Wilcoxon ($\alpha = 0.05$) en la comparación entre el algoritmo Alatasetal y MOPNAR

Medidas de Interés	**Comparación**	R^+	R^-	**Hipótesis**	p-**valor**
FC	MOPNAR vs. Alatasetal	340.5	10.5	**Rechazada**	<0.0001
Netconf	MOPNAR vs. Alatasetal	348	3	**Rechazada**	<0.0001
Yule'sQ	MOPNAR vs. Alatasetal	350	1	**Rechazada**	<0.0001
Lift	MOPNAR vs. Alatasetal	337	14	**Rechazada**	<0.0001

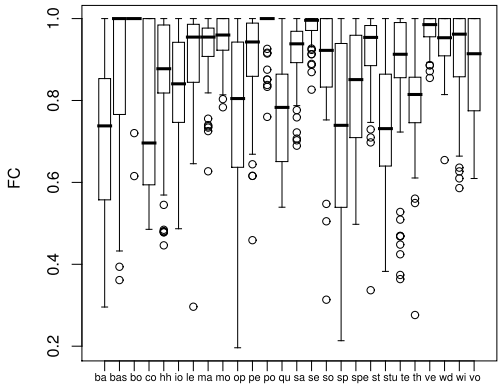

Figura 3.4: Boxplot de la medida FC para MOPNAR en todas las BDs

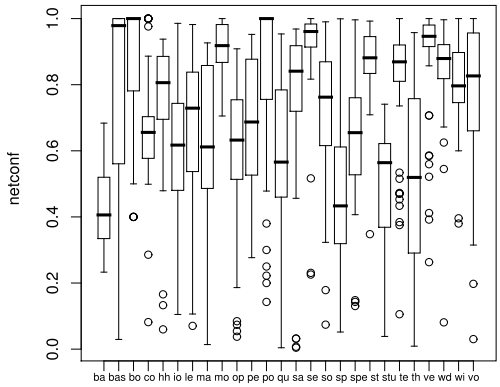

Figura 3.5: Boxplot de la medida netconf para MOPNAR en todas las BDs

se aprecia que la hipótesis de igualdad es rechazada en todos los casos obteniendo p-valores inferiores a 0.0001, mostrando claramente un mejor comportamiento de nuestra propuesta.

La Figura 3.4 y la Figura 3.5 representan boxplots que muestran los valores para las medidas FC y netconf, respectivamente, de las reglas obtenidas a partir de una de las 5 ejecuciones realizadas por nuestra propuesta para todas las BDs, seleccionada al azar. Podemos ver cómo todas las reglas representan dependencias positivas con valores cercanos al máximo valor de estas medidas (ver subsección 1.2.1 del capítulo 1). Observe que más del 75 % de las reglas obtenidas tienen un valor mayor que 0.5 para el FC y 0.3 para la medida netconf. La Figura 3.6 muestra los valores obtenidos por Alatasetal y nuestra propuesta para las medidas FC y netconf en la BD stock. Podemos ver que MOPNAR presenta mejores valores de FC y netconf que Alatasetal y todas sus reglas obtienen valores próximos a los

valores máximos que estas medidas pueden alcanzar. Además, se aprecia cómo algunas reglas obtenidas por Alatasetal representan independencia o dependencia negativa según estas medidas.

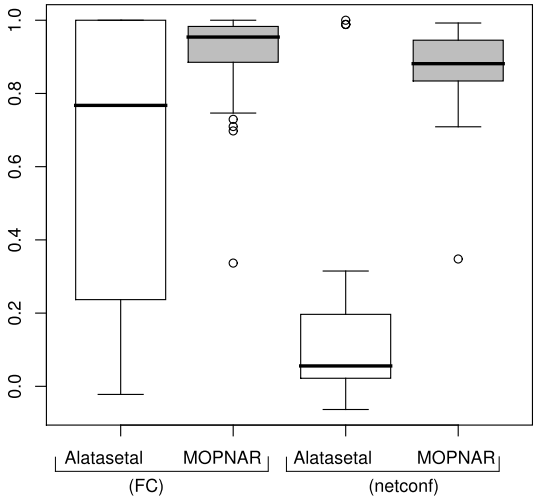

Figura 3.6: Boxplot de las medidas FC y netconf para el algoritmo Alatasetal y MOPNAR en la BD stock

3.2.3. Comparación con otros algoritmos evolutivos

En esta sección comparamos el rendimiento de nuestra propuesta con tres algoritmos genéticos mono-objetivo (EARMGA [YZZ09], GAR [MAR02] y GENAR [MAR01]) y tres AEMOs para extraer RACs (ARMMGA [QNMB11], MODENAR [AA08], MOEA_Ghosh [GN04]). La Tabla 3.4 muestra los resultados medios obtenidos por los métodos en todas las BD (la cabecera de esta tabla ha sido introducida en la sección anterior). Los resultados obtenidos por cada método en cada BD se pueden encontrar en la Tablas C.8, C.9, C.10 y C.11 de la sección C.2 del apéndice C. A partir del análisis de los resultados presentados en estas tablas podemos destacar los siguientes aspectos:

- Las reglas obtenidas por nuestra propuesta presentan valores de las medidas de interés mejores o similares a las reglas obtenidas por los algoritmos analizados en todas las BDs. Al igual que con Alatasetal, algunos de estos

Tabla 3.4: Resultados del valor medio de las medidas para todas las BDs en la comparación entre MOPNAR y los métodos evolutivos

Algoritmos	*#R*	Med_{Sop}	Med_{Conf}	Med_{Lift}	Med_{Conv}	Med_{FC}	$Med_{Netconf}$	$Med_{Yule'sQ}$	Med_{Amp}	*%Reg*
EARMGA	93	0,33	**1**	1,16	∞	0,09	0,01	0,04	**2,03**	99,2
GAR	52,1	0,56	0,88	1,45	∞	0,42	0,33	0,58	2,10	84,81
GENAR	29,46	0,24	0,83	3,38	∞	0,57	0,39	0,66	28,96	65,97
ARMMGA	1	**0,66**	0,91	1,34	∞	0,46	0,35	0,62	**2,03**	66,36
MODENAR	48,52	0,31	0,80	**187,31**	∞	0,53	0,27	0,56	16,82	62,17
MOEA_Ghosh	**150,13**	0,39	0,61	133,529	∞	0,31	0,19	0,35	23,43	55,61
MOPNAR	70,98	0,28	0,91	12,39	∞	**0,87**	**0,7**	**0,96**	2,92	**99,49**

algoritmos obtienen buenos promedios de soporte y confianza, pero los valores promedio para el resto de las medidas son bajos en algunas BDs, debido a que obtienen reglas que tienen ítems con soporte alto en el consecuente o dependencias negativas.

- MOPNAR nos permite extraer conjuntos reducidos de RACPNs que nos proporcionan un conocimiento interesante de todas las BDs, presentando valores medios de cubrimiento de más del 99 % en todos los casos, mientras que el resto de los algoritmos analizados obtienen valores peores o similares a los obtenidos por MOPNAR. Observe que EARMGA obtiene el mejor cubrimiento para casi todas las BDs, pero sus reglas presentan valores bajos para las medidas de interés. Además, GENAR obtiene valores bajos de cubrimiento para casi todas las BDs porque las reglas obtenidas siempre involucran a todos los atributos de la BD.

- Finalmente, las reglas obtenidas por MOPNAR presentan un bajo número de atributos lo que facilita su interpretabilidad desde el punto de vista del usuario.

Para analizar los resultados obtenidos para las mismas medidas de interés FC, netconf, yule'sQ y lift hemos empleado los mismos tests estadísticos no paramétricos utilizados en la subsección 2.2.3 del capítulo 2. Utilizamos $\alpha = 0.05$ como el nivel de significación en todos los casos. Primeramente, aplicamos el test Friedman [Fri37] y el test de Iman-Davenport [ID80] para comprobar si existen diferencias significativas entre los algoritmos. La Tabla 3.5 muestra los resultados estadísticos de ambas pruebas, demostrando que existen diferencias significativas entre los algoritmos al obtener valores muy superiores a sus valores críticos asociados. La

Tabla 3.6 muestra los rankings de los diferentes algoritmos analizados en este estudio, calculados a través del test de Friedman, donde nuestra propuesta obtiene el mejor ranking para todas las medidas.

Tabla 3.5: Resultados del test de Friedman e Iman-Davenport ($\alpha = 0.05$) en la comparación entre los algoritmos evolutivos y MOPNAR

Test de Friedman				
	FC	**Netconf**	**Yule'sQ**	**Lift**
Valor Crítico	12.592	12.592	12.592	12.592
Estadística (X_F^2)	**73,66**	**81,95**	**94**	**100,66**
p **valor**	<0.0001	<0.0001	<0.0001	<0.0001
Test de Iman-Davenport				
	FC	**Netconf**	**Yule'sQ**	**Lift**
Valor Crítico	2.10	2.10	2.10	
Estadística (F_F)	**22,36**	**27,67**	**37,91**	**45,48**
p **valor**	<0.0001	<0.0001	<0.0001	<0.0001

Tabla 3.6: Ranking promedio de los algoritmos evolutivos en la comparación con MOPNAR

FC		**Netconf**		**Yule'sQ**		**Lift**	
Algoritmos	*Ranking*	*Algoritmos*	*Ranking*	*Algoritmos*	*Ranking*	*Algoritmos*	*Ranking*
EARMGA	6,32	EARMGA	6,55	EARMGA	6,65	EARMGA	6,15
MOEA_Ghosh	4,86	MOEA_Ghosh	4,8	MOEA_Ghosh	5,23	ARMMGA	5,95
GAR	4,38	MODENAR	3,92	MODENAR	3,98	GAR	5,25
ARMMGA	4,11	GAR	3,92	GAR	3,88	GENAR	3,86
MODENAR	3,46	ARMMGA	3,86	ARMMGA	3,84	MODENAR	2,54
GENAR	3,32	GENAR	3,63	GENAR	3,15	MOEA_Ghosh	2,13
MOPNAR	**1,51**	MOPNAR	**1,28**	MOPNAR	**1,25**	MOPNAR	**2,09**

Para comparar el mejor método según Friedman (MOPNAR) con el resto de métodos analizados aplicamos el test de Holm [Hol79] y el test de Finner [Fin93]. Los resultados de estas pruebas se presentan en la Tabla 3.7, donde los algoritmos se ordenan respecto al z-valor obtenido. La hipótesis de igualdad es rechazada con el resto de los métodos ($p < \alpha/i$) para las medidas FC, netconf y yule'sQ. Por otra parte, para la medida lift no se rechazada la hipótesis respecto a los AEMOs MODENAR y MOEA_Ghosh debido a que esta medida no tiene limite superior y estos métodos obtienen valores muy altos para algunas BDs. Además, nuestra propuesta obtiene valores medios para lift superiores a los de MODENAR

Tabla 3.7: Resultados de los tests de Holm y Finner ($\alpha = 0.05$) en la comparación entre los algoritmos evolutivos y MOPNAR

i	Algoritmos	z	p	Holm	Finner	**Hipótesis**
			FC			
6	EARMGA	8.024259	0	0.008333	0.008512	**Rechazada**
5	MOEA-Ghosh	5.584884	0	0.01	0.016952	**Rechazada**
4	GAR	4.782459	0.000002	0.0125	0.025321	**Rechazada**
3	ARMMGA	4.3331	0.000015	0.016667	0.033617	**Rechazada**
2	MODENAR	3.241801	0.001188	0.025	0.041844	**Rechazada**
1	GENAR	3.017121	0.002552	0.05	0.05	**Rechazada**
			Netconf			
6	EARMGA	8.794588	0	0.008333	0.008512	**Rechazada**
5	MOEA-Ghosh	5.873758	0	0.01	0.016952	**Rechazada**
4	MODENAR	4.397294	0.000011	0.0125	0.025321	**Rechazada**
3	GAR	4.397294	0.000011	0.016667	0.033617	**Rechazada**
2	ARMMGA	4.301003	0.000017	0.025	0.041844	**Rechazada**
1	GENAR	3.915839	0.00009	0.05	0.05	**Rechazada**
			Yule'sQ			
6	EARMGA	9.019267	0	0.008333	0.008512	**Rechazada**
5	MOEA-Ghosh	6.644087	0	0.01	0.016952	**Rechazada**
4	MODENAR	4.557779	0.000005	0.0125	0.025321	**Rechazada**
3	GAR	4.397294	0.000011	0.016667	0.033617	**Rechazada**
2	ARMMGA	4.3331	0.000015	0.025	0.041844	**Rechazada**
1	GENAR	3.177607	0.001485	0.05	0.05	**Rechazada**
			Lift			
6	EARMGA	6.245875	0	0.008333	0.008512	**Rechazada**
5	ARMMGA	5.931837	0	0.01	0.016952	**Rechazada**
4	GAR	4.850149	0.000001	0.0125	0.025321	**Rechazada**
3	GENAR	2.721666	0.006495	0.016667	0.033617	**Rechazada**
2	MODENAR	0.697863	0.485263	0.025	0.041844	**No Rechazada**
1	MOEA-Ghosh	0.069786	0.944364	0.05	0.05	**No Rechazada**

y MOEA_Ghosh y en todas las BDs presenta valores superiores a 3, siendo un valor muy bueno para esta medida. Recordar que esta medida toma valor en el intervalo [0,∞), donde valores mayores que 1 representan dependencia positiva (ver subsección 1.2.1 del capítulo 1 para más información). Así, podemos concluir que MOPNAR obtiene el mejor rendimiento en comparación con el resto de los algoritmos analizados.

Las Figuras 3.7 y 3.8 son boxplots que muestran los valores para las medidas FC y netconf de las reglas obtenidas a partir de una de las 5 ejecuciones realizadas por todos los algoritmos evolutivos analizados en la BD stock, seleccionada al azar. Podemos ver que MOPNAR presenta los mejores valores de FC y netconf en comparación con el resto de los algoritmos, donde casi la totalidad de las reglas

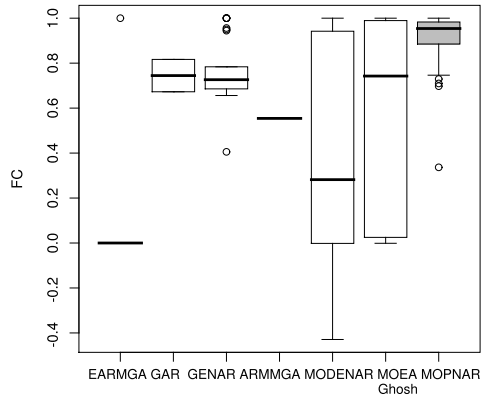

Figura 3.7: Boxplot de la medida FC para todos los algoritmos evolutivos en la BD stock en la comparación con MOPNAR

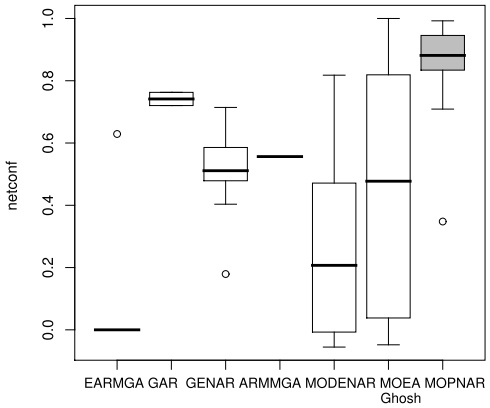

Figura 3.8: Boxplot de la medida netconf para todos los algoritmos evolutivos en la BD stock en la comparación con MOPNAR

obtienen valores superiores a 0.8 para estas medidas. Observe cómo algunas reglas obtenidas por MODENAR y EARMGA representan independencia o dependencia negativa según estas medidas.

3.2.4. Comparación con los algoritmos clásicos para extraer reglas de asociación

En esta sección comparamos el rendimiento de MOPNAR con los algoritmos clásicos para extraer reglas de asociación Apriori [SA96] y Eclat [Zak00], y con el AEMO clásico NSGA-II [DAPM02]. Para comparar nuestra propuesta con los métodos clásicos Apriori y Eclat, al igual que en la sección 2.2.4 del capítulo 2, hemos utilizado un particionamiento en anchura en cada atributo cuantitativo [LHTD02] y para compararnos con el AEMO NSGA-II, hemos utilizado el mismo esquema de codificación, objetivos, población inicial y operadores genéticos que en MOPNAR, extendiendo este AEMO a la extracción de reglas de asociación positivas y negativas.

La Tabla 3.8 muestra los resultados medios obtenidos por los métodos en todas las BD (la cabecera de esta tabla se describe en la subsección 3.2.2). Los resultados obtenidos por cada método en cada BD se pueden encontrar en las Tablas C.12 y C.13 de la sección C.2 del apéndice C. Observe que para los métodos Apriori

Tabla 3.8: Resultados del valor medio de las medidas para todas las BDs en la comparación entre MOPNAR y los métodos clásicos

Algoritmos	*#R*	Med_{Sop}	Med_{Conf}	Med_{Lift}	Med_{Conv}	Med_{FC}	$Med_{Netconf}$	$Med_{Yule'sQ}$	Med_{Amp}	*%Ejem*
Apriori	**8345221,46**	0,16	**0,93**	4,09	∞	0,81	0,64	0,84	5,13	90,43
Eclat	**8345221,46**	0,16	**0,93**	4,09	∞	0,8	0,64	0,84	5,13	90,43
NSGA-II	81,67	0,26	0,92	**66,96**	∞	**0,88**	0,69	0,94	3,2	89,7
MOPNAR	70,98	**0,28**	0,91	12,39	∞	0,87	**0,7**	**0,96**	**2,92**	**99,49**

Tabla 3.9: Resultados del test de Wilcoxon ($\alpha = 0{,}05$) en la comparación entre los algoritmos clásicos y MOPNAR

Comparación	R^+	R^-	**Hipótesis**	p**-valor**
		FC		
MOPNAR vs. Apriori	82	38	**No Rechazada**	$\geq 0{,}2$
MOPNAR vs. Eclat	82	38	**No Rechazada**	$\geq 0{,}2$
MOPNAR vs. NSGA-II	158	193	**No Rechazada**	$\geq 0{,}2$
		Netconf		
MOPNAR vs. Apriori	78,5	41,5	**No Rechazada**	$\geq 0{,}2$
MOPNAR vs. Eclat	78,5	41,5	**No Rechazada**	$\geq 0{,}2$
MOPNAR vs. NSGA-II	238,5	112,5	**No Rechazada**	0,11
		Yule'sQ		
MOPNAR vs. Apriori	86,5	18,5	**Rechazada**	0,03
MOPNAR vs. Eclat	86,5	18,5	**Rechazada**	0,03
MOPNAR vs. NSGA-II	300,5	50,5	**Rechazada**	$<$0.0001
		Lift		
MOPNAR vs. Apriori	107	13	**Rechazada**	0,005
MOPNAR vs. Eclat	107	13	**Rechazada**	0,005
MOPNAR vs. NSGA-II	37	314	**No Rechazada**	$\geq$ 0.2

y Eclat sólo mostramos los resultados de 15 BDs, debido a los problemas de escalabilidad que presentan, por lo que no pueden ejecutarse en todas las BDs. Para analizar los resultados obtenidos por las medidas de calidad hemos utilizado el test de Wilcoxon [She03, Wil45], considerando las medidas FC, netconf, yule'sQ y lift. La Tabla 3.9 muestra los resultados del test de Wilcoxon.

Observando los resultados del test de Wilcoxon, podemos ver que nuestra propuesta solo obtiene diferencias significativas en 2 de las 4 medidas (Yules'Q y Lift) con los algoritmos clásicos Apriori y Eclat al solo poder analizar los resultados de las 15 BDs donde estos algoritmos pudieron ejecutarse. Sin embargo, MOPNAR obtiene mejores rankings en todas las medidas de interés, con valores muy altos para estas medidas. Además, podemos ver en la Tabla 3.8 como Apriori y Eclat

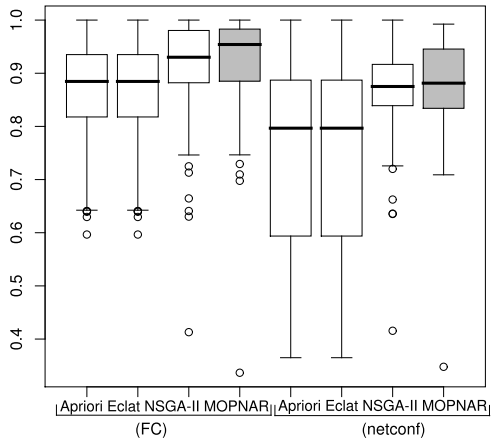

Figura 3.9: Boxplot de las medidas FC y netconf para los algoritmos clásicos y MOPNAR en la BD stock

extraen grandes conjuntos de reglas de asociación, siendo prácticamente imposibles de analizar por los usuarios. Por el contrario MOPNAR permite obtener un conjunto reducido de reglas con un alto cubrimiento en todas las BDs.

Los resultados estadísticos del test de Wilcoxon también muestran como MOPNAR solo tienen diferencias significativas en Yules'Q con el AEMO NSGA-II (obteniendo mejor ranking cada uno en dos medidas). Lo que resulta lógico al haber utilizado para NSGA-II los mismos componentes (codificación, objetivos, población inicial y operadores genéticos) que en nuestra propuesta. Sin embargo, podemos ver en la Tabla 3.8 como el modelo evolutivo MOEA/D-DE permite obtener conjuntos de reglas más reducidos y mejores cubrimientos de las BDs aunque NSGA-II también puede extraer reglas negativas. Esto pone de manifiesto las ventajas del modelo evolutivo MOEA/D-DE respecto al modelo de NSGA-II.

La Figura 3.9 representa un boxplot que muestra los valores para las medidas FC y netconf de las reglas obtenidas por los algoritmos clásicos y MOPNAR en la BD stock. Podemos ver que todas las reglas obtenidas por nuestra propuesta alcanzan valores cercanos al máximo valor de estas medidas, obteniendo reglas de gran calidad. Como hemos comentado, MOPNAR y NSGA-II obtienen reglas con valores para estas medidas muy similares al haber utilizado los mismos componentes. Por último, destacar que MOPNAR presenta mejores valores de FC y netconf que los algoritmos clásicos Apriori y Eclat, aunque en media sean cercanos y no permitan que hayan diferencias significativas.

3.2.5. Análisis de escalabilidad

Varios experimentos han sido realizados para analizar la escalabilidad de los algoritmos en la BD House_16H. Todos los experimentos se realizaron en un procesador Intel Core i7, 2,80 GHz CPU con 12 Gb de memoria y fueron ejecutados en Linux. El tiempo de ejecución promedio empleado por los algoritmos analizados cuando aumentan el número de atributos y registros se muestran en la Tabla 4.11 y Tabla 4.12, respectivamente.

Tabla 3.10: El tiempo de ejecución (segundos) empleado por todos los algoritmos comparados y MOPNAR cuando el número de atributos aumenta en la BD House_16H

	Número de Atributos				
Algoritmos	4	8	12	16	17
EARMGA	78	61	78	75	65
GAR	619	1528	1991	2083	2014
GENAR	24	38	**47**	57	**59**
Alatasetal	50	77	52	72	154
ARMMGA	104	93	95	100	97
MODENAR	60	81	97	95	93
MOEA_Ghosh	23	36	50	**46**	73
Apriori	**3**	**5**	233	5192	11268
Eclat	**3**	**5**	251	5812	12467
NSGA-II	54	56	55	96	77
MOPNAR	45	47	58	77	72

Las Figuras 3.10 y 3.11 muestran la relación entre el tiempo de ejecución y el número de atributos para todos los algoritmos estudiados. Podemos ver como casi todos los algoritmos evolutivos tienden a escalar linealmente cuando el número de atributos de la BD aumenta, sin embargo, como hemos comentado en la subsección 2.2.5 del capítulo 2 los algoritmos clásicos para extraer reglas de asociación (Apriori y Eclat) aumentan exponencialmente cuando el número de atributos es superior a 10.

Tabla 3.11: El tiempo de ejecución (segundos) empleado por todos los algoritmos comparados y MOPNAR cuando el número de registros aumenta en la BD House_16H

	Número de Registros				
Algoritmos	20 %	40 %	60 %	80 %	100 %
EARMGA	15	31	39	50	65
GAR	467	898	1260	1595	2014
GENAR	**12**	**23**	**36**	47	**59**
Alatasetal	16	24	93	104	154
ARMMGA	21	40	60	77	97
MODENAR	13	41	37	115	93
MOEA_Ghosh	13	24	37	**44**	73
Apriori	2689	5180	10050	9004	11268
Eclat	3076	8700	11300	10604	12467
NSGA-II	13	27	51	67	77
MOPNAR	16	31	47	59	72

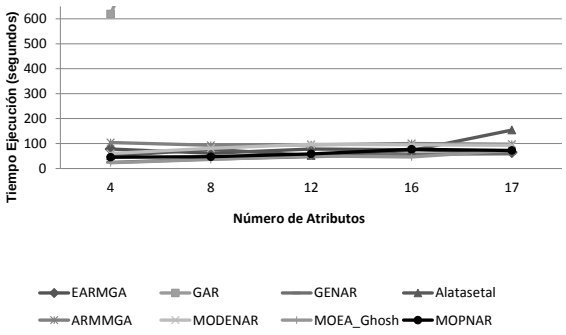

Figura 3.10: Relación entre el tiempo de ejecución y el número de atributos para los algoritmos evolutivos y MOPNAR en la BD House_16H

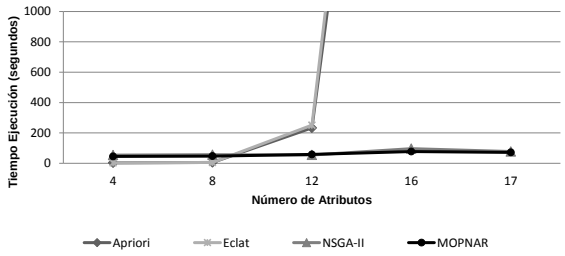

Figura 3.11: Relación entre el tiempo de ejecución y el número de atributos para los algoritmos clásicos y MOPNAR en la BD House_16H

Las Figuras 3.12 y 3.13 muestran la relación entre el tiempo de ejecución y el número de ejemplos. Al igual que en el caso anterior, el tiempo de ejecución tiende a escalar linealmente cuando el número de ejemplos en la BD aumenta. Por otra parte, podemos ver cómo el aumento en el número de registros y atributos afecta los algoritmos clásicos para extraer reglas de asociación (Apriori y Eclat) más que a los algoritmos evolutivos. Observe que la Figura 3.10 y la Figura 3.12 muestran pocos resultados sobre GAR, porque su tiempo de ejecución es superior a más de 650 segundos en casi todos los casos, lo que fácilmente se puede ver en la Tabla 4.11 y Tabla 4.12.

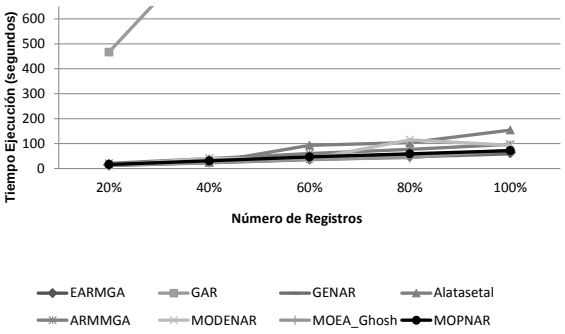

Figura 3.12: Relación entre el tiempo de ejecución y el número de registros para los algoritmos evolutivos y MOPNAR en la BD House_16H

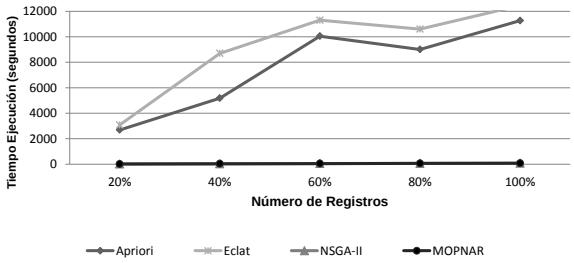

Figura 3.13: Relación entre el tiempo de ejecución y el número de registros para los algoritmos clásicos y MOPNAR en la BD House_16H

3.2.6. Reglas obtenidas por nuestra propuesta

En esta sección se estudian algunas RACPNs obtenidas por nuestra propuesta. La Tabla 3.12 muestra varias RACPNs interesantes obtenidas de cuatro BDs, donde BD es la BD en la que se obtuvo la regla, la $Regla$ es la regla obtenida, Sop y $Conf$ son, respectivamente, el soporte y la confianza de las reglas y FC es el valor de la medida FC de las reglas.

Tabla 3.12: Reglas obtenidos por MOPNAR en varias BDs

BD	*Regla*	*Sop*	*Conf*	*FC*
Bolts	R1: Si **SENS** no está [7.0, 10.0] entonces **SPEED1** no está [3.0, 5.0]	0.49	1	1
Flare	R2: Si **HistComplex** no es 2 entonces **X-class** es 0	0.59	1	1
Quake	R3: Si **Latitude** está [-10.58, 47.44] entonces **Longitude** no está [-179.96, -155.45]	0.59	1	1
Stock	R4: Si **Company2** no está [46.38, 56.99] entonces **Company1** no está [17.22, 31.99]	0.58	0.99	0.97

Estas reglas pueden ser interpretadas de la siguiente manera. La BD Bolts fue generada para almacenar datos de diferentes experimentos sobre los efectos de los ajustes de una máquina para cortar tornillos a lo largo del tiempo. $R1$ indica que cuando la sensibilidad del ojo electrónico (SENS) es inferior a 7 (el dominio de esta variable es [0, 10]), entonces la velocidad de rotación (SPEED1) de la placa es menor que 3 (bajo) o superior a 5 (muy rápida). La BD Flare almacena los datos

sobre el número de veces que un determinado tipo de destello solar se produce en un período de 24 horas. $R2$ muestra que cuando la región no es históricamente compleja (HistComplex), entonces los destellos severos (clase X) no se producen en las siguientes 24 horas. La BD Quake proporciona los datos sobre métodos de suavizado para estadísticas en la predicción de terremotos. En este caso, $R3$ indica que cuando la latitud es entre -10.58 y 44.77 entonces la longitud es inferior a -155.45. Los datos proporcionados por la BD stock son precios de las acciones diarias de diez empresas aeroespaciales almacenados desde enero de 1988 hasta octubre de 1991. $R4$ muestra que cuando la segunda compañía (Company2) no tiene precios entre 46.38 y 56.99 entonces la primera compañía (Company1) no tiene precios entre 17.22 y 31.99.

La mayoría de las reglas obtenidas por nuestra propuesta incluyen ítems negativos, lo que nos permite reducir el número de reglas necesarias para extraer conocimiento interesante de las BDs. La Tabla 3.13 muestra algunas RACPNs obtenidas por nuestra propuesta y las RACs positivas que necesitaron obtener otros algoritmos analizados para extraer el mismo conocimiento, donde BD es la BD en la que se obtuvieron las reglas, *Nuestra Propuesta* representa las RACPNs obtenidas por nuestra propuesta y *Otros algoritmos* representa las RACs positivas obtenidas por otros algoritmos.

Tabla 3.13: Relación entre RACPNs obtenidas por MOPNAR y RACs positivas obtenidos por otros algoritmos

BD	*Nuestra Propuesta*	*Otros algoritmos*
Bolts	Si **SENS** está [0, 6] entonces **SPEED2** no está [1.5, 2.4]	Si **SENS** está [0, 6] entonces **SPEED2** está [0, 1.5] Si **SENS** está [0, 6] entonces **SPEED2** está [2.5, 2.5]
Quake	Si **Focal depth** no está [136, 176] y Si **Longitude** está [-179.8, -171.1] entonces **Latitude** está [-33.7, -14.9]	Si **Focal depth** está [0, 135] y **Longitude** está [-179.8, -171.1] entonces **Latitude** está [-33.7, -14.9] Si **Focal depth** está [177, 656] y **Longitude** está [-179.8, -171.1] entonces **Latitude** está [-33.7, -14.9]
Stock	Si **Company2** no está [46.3, 56.9] entonces **Company1** no está [17.2, 31.9]	Si **Company2** está [19.2, 46.3] entonces **Company1** está [32, 61.5] Si **Company2** está [57, 60.2] entonces **Company1** está [32, 61.5]

3.3. Sumario

En este capítulo hemos propuesto MOPNAR, un nuevo AEMO que extiende el AEMO basado en descomposición MOEA/D-DE [LZ09] para extraer un conjunto reducido de RACPNs que son fáciles de entender, interesantes y con un buen cubrimiento de la BD. Nuestra propuesta realiza un aprendizaje evolutivo de los intervalos de los atributos y una selección de las condiciones para cada regla, maximizando tres objetivos: rendimiento, interés y comprensibilidad. MOPNAR introduce un proceso de reinicialización y una PE al modelo evolutivo, con el fin de promover diversidad en la población, almacenar todas las reglas no dominadas encontradas y mejorar el cubrimiento de la BD. Además, las reglas obtenidas son muy fuertes, lo que indica una fuerte dependencia entre los ítems y resuelve las limitaciones del soporte. A partir del estudio realizado podemos concluir:

- Además, MOPNAR obtiene conjuntos reducidos de RACPNs con pocos atributos, lo que permite una mejor comprensión del usuario, y con valores altos para las medidas de interés en todas las BDs.
- La extracción no sólo de reglas positivas sino también de reglas negativas, junto con la utilización del modelo evolutivo MOEA/D-DE, nos permite reducir el número de reglas necesarias para extraer conocimiento interesante de las BDs, obteniendo conjuntos reducidos de reglas de buena calidad con un buen equilibrio entre el número de reglas, el soporte y el cubrimiento de la BD.
- Por último, el enfoque propuesto presenta un buen coste computacional en todas las BDs y una buena escalabilidad cuando el tamaño del problema aumenta.

Capítulo 4

NIGAR: Algoritmo Genético basado en Nichos para Extraer un Conjunto Interesante y Diverso de Reglas de Asociación Cuantitativas

Como hemos visto en el capítulo 1 los algoritmos genéticos han sido ampliamente utilizados en la extracción de RACs. Sin embargo, estos algoritmos obtienen reglas muy similares porque tienden a converger a la mejor solución, afectando a la diversidad del conjunto de reglas obtenido. Por esto, la búsqueda y el mantenimiento de múltiples soluciones en la población es un reto para el uso de los algoritmos genéticos en problemas como el de extracción de reglas de asociación, problemas con varios óptimos globales (altamente multi-modales) donde todos las soluciones óptimas deben ser obtenidas. Los métodos basados en nichos (ver sección 1.4.3 del capítulo 1) extienden los algoritmos genéticos para permitirles localizar y mantener múltiples soluciones globales en la población evitando la convergencia hacia una sola solución. Los AGNs pueden por lo tanto ser un método interesante para abordar el problema de la extracción de de reglas de asociación, proporcionando todas las reglas de alta calidad (los óptimos globales

del problema) que representan un conocimiento interesante y diferente entre sí de la BD.

Por ello, en este capítulo presentamos un nuevo AGN, llamado NIGAR, para extraer un conjunto diverso y reducido de reglas de asociación positivas y negativas fáciles de entender, interesantes y con un buen cubrimiento de la BD. Nuestra propuesta realiza un aprendizaje evolutivo de las reglas haciendo uso de un mecanismo de penalización, un proceso de reinicialización y una PE para gestionar los nichos en la población. Esta propuesta introduce dos valores umbral que nos permiten ajustar el equilibrio entre la diversidad y la calidad de las reglas obtenidas. Además, presentamos una nueva medida de distancia basada en dos componentes de las reglas: el ratio de solapamiento de los atributos comunes y los ejemplos cubiertos por las dos reglas.

El capítulo se organiza en las siguientes secciones. La sección 4.1 describe el algoritmo evolutivo propuesto para extraer un conjunto diverso y reducido de RACPNs. En la sección 4.2 se muestra el estudio experimental realizado sobre 26 BDs reales para evaluar la efectividad de nuestra propuesta. Por último, se presenta un breve resumen del capítulo en la sección 4.3.

4.1. Algoritmo Genético basado en nichos para extraer Reglas de Asociación Cuantitativas Positivas y Negativas: NIGAR

Esta sección presenta nuestra propuesta para obtener un conjunto diverso de RACPNs con un buen equilibrio entre calidad y diversidad del conocimiento obtenido. Esta propuesta realiza un aprendizaje evolutivo de las reglas y combina un mecanismo de penalización, una PE y un proceso de reinicialización para mantener múltiples soluciones óptimas en la población e introducir diversidad en el proceso de búsqueda. A continuación, veremos en detalle todas sus características.

4.1.1. Gestión de nichos dentro de NIGAR y nueva medida de distancia

En esta propuesta consideramos el uso de un AGN para realizar un aprendizaje evolutivo de las reglas, el cual extiende a los algoritmos genéticos para localizar

y mantener múltiples soluciones en la población y evitar la convergencia hacia una sola solución. Para gestionar los nichos en la población y promover diversidad en la población, hemos introducido una PE, un mecanismo de penalización y un proceso de reinicialización en el proceso de búsqueda. La PE nos va a permitir que no se pierda la mejor solución de ninguno de los nichos encontrados durante el proceso de búsqueda, manteniendo la mejor solución de cada uno de los nichos generados. Además, con el objetivo de promover la búsqueda de soluciones óptimas distintas a las que tenemos en la PE, el mecanismo de penalización penalizará a los individuos que pertenezcan al mismo nicho que alguna de las soluciones de la PE y su valor para la función objetivo sea menor que el de la mejor solución de su nicho. De esta forma se fomenta en el proceso de búsqueda el desarrollo de nichos en otras zonas del espacio de búsqueda en las que todavía no se haya encontrado un óptimo del problema.

Una vez que el número de soluciones nuevas en la población sea menor que un $\alpha\,\%$ del tamaño de la población actual, se reinicia la población para buscar en otras zonas del espacio de búsqueda y se actualiza la PE con la mejor solución de cada nicho encontrado en la población, comprobando que su calidad sea superior a un $Ev_{Min}\,\%$ de la media del valor para la función objetivo de los individuos de la PE. El umbral Ev_{Min} permitirá al experto evitar la entrada de soluciones de baja calidad en la PE, las cuales se corresponderán con óptimos locales del problema. De esta manera, se mantiene un mínimo de calidad de las reglas que proporcionará nuestro método al final del proceso. Para evitar redundancia en las reglas de la PE, se comprueba si en la PE existen reglas que pertenezcan al mismo nicho, en cuyo caso nos quedaremos con la mejor regla del nicho.

Para determinar si dos soluciones pertenecen a un mismo nicho se analizan dos componentes de las reglas, el ratio de solapamiento de los atributos comunes y los ejemplos cubiertos por las dos reglas. Si el ratio de solapamiento de los atributos comunes de las reglas supera un valor umbral $Nich_{Min}$ ($Nich_{Min}$ definido por el usuario), se considera que el conocimiento representado por las dos reglas es parecido y pasaremos a comprobar si este conocimiento lo están ofreciendo sobre los mismos ejemplos de la BD. Para ello, se calcula el ratio de ejemplos cubiertos por las dos reglas y si este también supera el umbral $Nich_{Min}$ definido por el usuario, las dos reglas pertenecerán al mismo nicho al proporcionar un conocimiento similar sobre el problema. Destacar que el umbral $Nich_{Min}$ nos va a permitir gestionar cuán fácil o difícil será agrupar soluciones en un mismo nicho.

El solapamiento entre dos atributos A_1 y A_2 es el valor más grande que resulta de dividir la amplitud de la parte que se solapan los dos intervalos entre la

amplitud de cada uno de los intervalos. Para los atributos nominales, si los valores de ambos atributos son iguales el solapamiento es 1 en caso contrario será 0. Destacar que para el caso de los atributos negativos se calcula el solapamiento considerando su intervalo positivo equivalente. Por lo tanto, el solapamiento entre dos atributos A_1 y A_2 se define como:

$$solap\,(A_1, A_2) = Max\left(\frac{Amplitud_{Comun}}{Amplitud_{A_1}}, \frac{Amplitud_{Comun}}{Amplitud_{A_2}}\right) \tag{4.1}$$

Esta medida toma valores en el intervalo [0,1], donde valores cercanos a 0 representan que los atributos son muy diferentes, ya que presentan un bajo solapamiento entre los intervalos y valores cercanos a 1 representan que los atributos son muy similares al presentar un alto grado de coincidencia entre los intervalos. Teniendo en cuenta esta definición de solapamiento, el ratio de solapamiento entre las reglas R_1 y R_2 se define como:

$$ratio_{solap}(R_1, R_2) = \frac{\sum_{i=1}^{CA} solap(A_{i_{R_1}}, A_{i_{R_2}})}{\mid CA \mid} \tag{4.2}$$

donde CA son los atributos comunes de las dos reglas y $\mid CA \mid$ es el número de atributos comunes de las dos reglas. Destacar que esta medida solo considera los atributos comunes entre las reglas porque si dos reglas presentan atributos diferentes, estas deberían pertenecer a nichos distintos al proporcionar un conocimiento diferente sobre el problema, aunque cubran los mismos ejemplos.

El ratio de ejemplos cubiertos por dos reglas se define como el mayor valor obtenido de dividir la cantidad de ejemplos cubiertos por ambas reglas entre la cantidad de ejemplos q cubren cada una de ellas, es decir, representa que tanto porciento representan los ejemplos comunes respecto a los ejemplos que cubren cada una de las reglas. Esta medida toma valores en el intervalo [0,1], donde valores cercanos a 0 representan que las reglas no tienen muchos ejemplos comunes y valores cercanos a 1 representan que las reglas prácticamente cubren los mismos ejemplos. El ratio de ejemplos cubiertos se define como:

$$ratio_{cub}(R_1, R_2) = Max\left(\frac{cub(R_1R_2)}{cub(R_1)}, \frac{cub(R_1R_2)}{cub(R_2)}\right) \tag{4.3}$$

donde $cub(R_1R_2)$ representa el número de ejemplos cubiertos por ambas reglas R_1 y R_2, y $cub(R_1)$ y $cub(R_2)$ representan el número de ejemplos cubiertos por R_1 y R_2 respectivamente. Destacar que para las reglas que son especializaciones de otra regla, esta medida obtendrá su máximo valor.

Tabla 4.1: BD simple con ocho ejemplos usada para calcular la distancia entre dos reglas

ID	X_1	X_2	X_3
ID1	20	1,5	7
ID2	19,5	1	6
ID3	20	2,1	6
ID4	22	4,3	7
ID5	30	8,4	13
ID6	20	2,4	2
ID7	24	9	5
ID8	23	5,2	10

Por ejemplo, consideremos una BD simple con tres atributos X_1, X_2 y X_3, 8 ejemplos y $Nich_{Min} = 0{,}5$. La Tabla 4.1 muestra los valores de los 8 ejemplos de la BD. Además, supongamos que tenemos las reglas: R_1: $X_1 \in [20, 20]$ y $X_2 \in [1`5, 3`5] \rightarrow X_3 \in [1, 8]$ y R_2: $X_1 \in [18, 25] \rightarrow X_3 \in [5, 14]$. Para ver si estas dos reglas pertenecen al mismo nicho, primero calculamos el ratio de solapamiento entre las reglas de la siguiente manera:

$$solap\left(X_{1_{R_1}}, X_{1_{R_2}}\right) = 1$$

$$solap\left(X_{3_{R_1}}, X_{3_{R_2}}\right) = Max\left(\frac{3}{7}, \frac{3}{9}\right) = 0,42$$

$$ratio_{solap}(R_1, R_2) = \frac{1 + 0,42}{2} = 0,71$$

Como el ratio de solapamiento de los atributos comunes es mayor que el umbral $Nich_{Min}$, comprobamos si el ratio de ejemplos cubiertos entre las dos reglas supera también el umbral $Nich_{Min}$. La Tabla 4.1 muestra como R_1 cubre los ejemplos ID1, ID3 y ID6, y R_2 cubre ID1, ID2, ID3, ID4, ID7, ID8, luego los ejemplos comunes son: ID1 y ID3. Teniendo en cuenta esta situación, el ratio de ejemplos cubiertos se calcula como:

$$ratio_{cub}(R_1, R_2) = Max\left(\frac{2}{3}, \frac{2}{6}\right) = 0,66$$

Como el ratio de ejemplos cubiertos también supera el umbral $Nich_{Min}$, las reglas R_1 y R_2 serían agrupadas en el mismo nicho.

A partir de estas características hemos propuesto una nueva medida de distancia entre reglas basada en estos dos componentes de la reglas: el ratio de solapamiento de los atributos comunes y los ejemplos cubiertos por las dos reglas. Esta medida se define como:

$$acir(R_1, R_2) = 1 - \frac{ratio_{solap}(R_1, R_2) + ratio_{cub}(R_1, R_2)}{2} \tag{4.4}$$

Esta medida es simétrica y toma valores en el intervalo [0, 1], donde valores cercanos a 0 representan reglas muy similares y cercanos a 1 reglas muy distintas. Destacar que esta medida nos permite medir como de peculiares son las reglas (ver subsección 1.2.1 del capítulo 1), al tener en cuenta la similitud de los atributos comunes que involucran y los ejemplos comunes que cubren con otras reglas.

Esta medida de distancia es utilizada por el mecanismo de penalización para graduar la penalización del valor para la función objetivo de los individuos de la población que pertenecen al mismo nicho que alguna de las soluciones de la PE. Así el valor de la función objetivo para estas soluciones será modificado de la siguiente forma:

$$F'_C(C) = F_C(C) - ((1 - acir(C, Sol_{EP})) * (F_C(C) * 0,2)) \tag{4.5}$$

donde F_C es el valor de la evaluación del cromosoma en la función objetivo (ver subsección 4.1.3), y Sol_{EP} es la solución del nicho más cercano de la PE, es decir, su regla más similar de la PE, ya que cada regla en la PE es un nicho. Destacar que el valor de la función objetivo para una solución será penalizado con un valor máximo de un χ % de su valor para la función objetivo (en esta propuesta hemos utilizado el 20 %, es decir 0,2).

4.1.2. Esquema de Codificación y Población inicial

El esquema de codificación utilizado en esta propuesta es el mismo que se describió en la subsección 3.1.2 del capítulo 3. Cada cromosoma es un vector de n genes que representa los atributos e intervalos de una regla, donde n es el número de atributos de la BD. Cada uno de los n genes estará compuesto por 4 partes: *ac* indica si un gen es considerado en la regla, si es parte del antecedente, del consecuente o no está involucrado en la regla; *pn* indica si el intervalo es

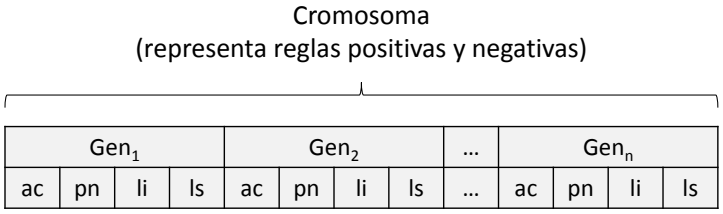

Figura 4.1: Esquema de un cromosoma para codificar reglas positivas y negativas

positivo o negativo; *li* y *ls* representan el límite inferior y superior del intervalo respectivamente. La Figura 4.1 muestra el esquema de este cromosoma.

En esta propuesta también hemos utilizado *Amplitud* (vea ecuación 2.1 del capítulo 2) para evitar que los intervalos crezcan hasta cubrir el dominio completo de los atributos. Así, ningún intervalo positivo podrá tener un tamaño mayor que *Amplitud* y ningún intervalo negativo podrá tener un tamaño menor que *Amplitud*.

Para crear la población inicial se ha utilizado el proceso descrito en la subsección 3.1.2 del capítulo 3. En este proceso se crea una población inicial con reglas que presentan un solo atributo en el consecuente (aunque el esquema permite codificar reglas con más de un atributo en el consecuente) y un buen cubrimiento de la BD. Para ello, primero se seleccionan aleatoriamente los atributos que formarán parte del antecedente y del consecuente. Después se selecciona si el intervalo será positivo o negativo. Luego, se selecciona aleatoriamente un ejemplo de la BD para generar los intervalos de cada atributo. Un intervalo se crea con un tamaño igual al 50 % del valor de *Amplitud* del atributo correspondiente, donde el valor del ejemplo seleccionado quedará en el centro del intervalo generado. Si algún intervalo supera el límite del dominio de su atributo, entonces se reemplaza el límite del intervalo por el límite del dominio del atributo. Finalmente se marcan los ejemplos que cubre la regla generada en la BD con el fin de utilizar los ejemplos que no han sido cubiertos para generar las próximas reglas. Este proceso se repite para todos los ejemplos que no han sido marcados hasta que se complete la población inicial. Si todos los ejemplos se han marcado y la población inicial no se ha completado, se vuelven a desmarcar todos los ejemplos y el proceso se repite hasta que la población inicial sea completada. Por último, destacar que la PE inicialmente está vacía.

4.1.3. Evaluación de los Cromosomas

Para evaluar un cromosoma se utiliza una función de evaluación que maximiza la agregación de tres métricas de las reglas. Esta medida toma valores entre -1 y 2,5, donde -1 representa el peor valor y 2,5 el mejor valor. Esta función se define como sigue:

$$F_C(C) = metrica_1 + metrica_2 + metrica_3 \tag{4.6}$$

La primera métrica combina el soporte y la medida de interés lift, favoreciendo las reglas con mejor cubrimiento pero que también sean interesantes. Destacar que si una regla tiene un gran cubrimiento pero su interés es muy bajo, el valor de esta medida para esta regla será bajo. Esta métrica toma valores en el intervalo [0,1] y se define como:

$$metrica_1 = \left(1 - \frac{1}{2^{10 * soporte(A \to B)}}\right) * \left(1 - \frac{1}{lift(A \to B)}\right) \tag{4.7}$$

La segunda métrica mide como de interesante es una regla para el usuario. En esta propuesta hemos utilizado la medida netconf (ver subsección 1.2.1 del capítulo 1), la cual nos permite detectar dependencia negativa, positiva o independencia entre los ítems.

Finalmente, la tercera métrica mide el número de atributos que contiene la regla, porque las reglas cuando involucran muchos atributos pueden resultar difíciles de comprender. Esta métrica toma valores entre 0 y 1, los cuales representan el peor y el mejor valor, respectivamente. Esta se define de la siguiente manera, donde $Atrib_{X \to Y}$ representa solo los atributos involucrados en el antecedente de la regla, porque en esta propuesta solo consideramos reglas con un solo atributo en el consecuente.

$$Comprensibilidad(X \to Y) = 1/Atrib_{X \to Y} * 2 \tag{4.8}$$

Destacar que solo nos interesan las reglas de asociación fuertes [BBSV02] porque indican dependencias positivas entre los ítems y evitan el problema que presentan los ítems con alto soporte (ver sección 1.2.2 del capítulo 1). Por esto, a las reglas que no sean fuertes se les asignará el peor valor de la función objetivo para que que sean eliminadas de la población.

4.1.4. Operadores genéticos

Para obtener dos hijos a partir de dos padres seleccionados utilizando una selección por torneo binario, el operador de cruce genera dos hijos intercambiando aleatoriamente los genes de los dos padres (exploración). Además, el operador PCBLX [LHKM04] (un operador basado en el operador BLX-α [ES93]) es aplicado sobre los límites de los intervalos de los atributos numéricos. Este tipo de operadores está basado en el concepto de entornos, permitiéndonos generar los límites de los intervalos de los hijos alrededor de los límites de los intervalos de los padres. La Figura 4.2 representa el comportamiento de este operador.

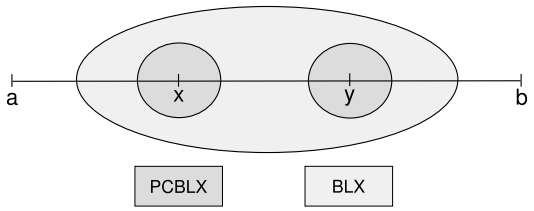

Figura 4.2: Esquema del comportamiento de los operadores BLX y PCBLX

Por ejemplo, supongamos que $X = [l_{i_x}, l_{s_x}]$ y $Y = [l_{i_y}, l_{s_y}]$, donde l_{i_x}, l_{s_x}, l_{i_y}, $l_{s_y} \in [a_i, b_i] \subset \Re$, son dos intervalos de un atributo numérico i que van a ser cruzados. Los límites de los hijos $[O_1^{l_i}, O_1^{l_s}]$ y $[O_2^{l_i}, O_2^{l_s}]$, se generan de la siguiente manera:

- $O_1^{l_i}$ = es elegido aleatoriamente (uniformemente) dentro del intervalo $[O_{1_{min}}^{l_i}, O_{1_{max}}^{l_i}]$, con $O_{1_{min}}^{l_i} = \max\{a, l_{i_x} - I^{l_i} \cdot\alpha\}$, $O_{1_{max}}^{l_i} = \min\{b, l_{i_x} + I^{l_i} \cdot\alpha\}$, y $I^{l_i} = \mid l_{i_x} - l_{i_y} \mid$.

- $O_2^{l_i}$ = es elegido aleatoriamente (uniformemente) dentro del intervalo $[O_{2_{min}}^{l_i}, O_{2_{max}}^{l_i}]$, con $O_{2_{min}}^{l_i} = \max\{a, l_{i_y} - I^{l_i} \cdot\alpha\}$, $O_{2_{max}}^{l_i} = \min\{b, l_{i_y} + I^{l_i} \cdot\alpha\}$.

- $O_1^{l_s}$ = es elegido aleatoriamente (uniformemente) dentro del intervalo $[O_{1_{min}}^{l_s}, O_{1_{max}}^{l_s}]$, con $O_{1_{min}}^{l_s} = \max\{a, l_{s_x} - I^{l_s} \cdot\alpha\}$, $O_{1_{max}}^{l_s} = \min\{b, l_{s_x} + I^{l_s} \cdot\alpha\}$, and $I^{l_s} = \mid l_{s_x} - l_{s_y} \mid$.

- $O_2^{l_s}$ = es elegido aleatoriamente (uniformemente) dentro del intervalo $[O_{2_{min}}^{l_s}, O_{2_{max}}^{l_s}]$, con $O_{2_{min}}^{l_s} = \max\{a, l_{s_y} - I^{l_s} \cdot \alpha\}$, $O_{2_{max}}^{l_s} = \min\{b, l_{s_y} + I^{l_s} \cdot \alpha\}$.

Una vez que el operador de cruce ha generado los hijos aplicamos sobre ellos los mismos operadores de mutación y reparación utilizados para nuestra propuesta MOPNAR (ver subsección 3.1.4 del capítulo 3). El operador de mutación modificará aleatoriamente los componentes de un gen seleccionado al azar. Primero selecciona el valor de *ac* aleatoriamente, dentro del conjunto {-1,0,1} y luego el valor de *pn* dentro del conjunto {0,1}. Luego selecciona al azar uno de los límites del intervalo y aumenta o disminuye su valor de manera aleatoria. Para disminuir el intervalo se seleccionará al azar un nuevo valor entre el valor actual del límite y el valor del otro límite del intervalo y si se va a aumentar el intervalo, entonces el valor será seleccionado entre el valor actual del límite y el límite del dominio del atributo.

El operador de reparación evita que tengamos en la población reglas que tengan más de un atributo en el consecuente o que no tengan ningún atributo en el antecedente o consecuente. Además, para obtener reglas más simples este operador decrementa el tamaño de los intervalos positivos mientras que se cubran los mismos ejemplos cubiertos por los intervalos originales. En el caso de los intervalos negativos el tamaño de los intervalos será incrementado, reduciendo el dominio que cubren (ver subsección 3.1.4 del capítulo 3 para una descripción detallada del operador).

4.1.5. Proceso de Reinicialización

Para promover diversidad en la búsqueda se aplica el proceso de reinicialización, el cual nos permite localizar nichos en otras áreas del espacio de búsqueda en las que no hayamos encontrado ninguno. Este proceso se aplica cuando el número de nuevas soluciones en la población es menor que un $\alpha\,\%$ del tamaño de la población actual. En este proceso antes de reiniciar la población se actualiza la PE utilizando las mejores soluciones de los nichos de la población actual (ver la subsección 4.1.1 para una descripción más detallada). Luego, se marcan los ejemplos que hayan sido cubiertos por las reglas de la PE y se aplica nuevamente el proceso de inicialización sobre los ejemplos que no hayan sido marcados (ver subsección 4.1.2). Finalmente se aplica el mecanismo de penalización sobre los nuevos individuos creados en este proceso.

4.1.6. Modelo Evolutivo

El modelo evolutivo de nuestra propuesta es como sigue. Primero, generamos una población inicial con un conjunto de reglas que nos proporcionen un buen cubrimiento de la BD. Luego, se generan los hijos a partir de la población actual aplicando los operadores de cruzamiento, mutación y reparación. Después se aplica el mecanismo de penalización para cada individuo nuevo, penalizando su valor para la función objetivo si pertenece al mismo nicho que alguna de las soluciones de la PE y su valor para la función objetivo es menor que el de la mejor solución del nicho (ver subsección 4.1.1). La siguiente población estará formada por los mejores individuos entre padres e hijos. Finalmente, cuando el número de nuevos individuos es menor que un $\alpha\,\%$ del tamaño de la población actual se actualiza la PE y se aplica el proceso de reinicialización. Destacar que la PE se actualiza con la mejor solución de cada nicho encontrado en la población, comprobando que su calidad sea superior a un $Ev_{Min}\,\%$ de la media de la evaluación de los individuos de la PE. Por último, se comprueba si alguno de los individuos de la PE pertenecen al mismo nicho, en cuyo caso nos quedaremos sólo con la mejor solución del nicho. Todo este proceso se repite hasta que se cumpla la condición de parada.

La Figura 4.3 muestra un esquema del método evolutivo.

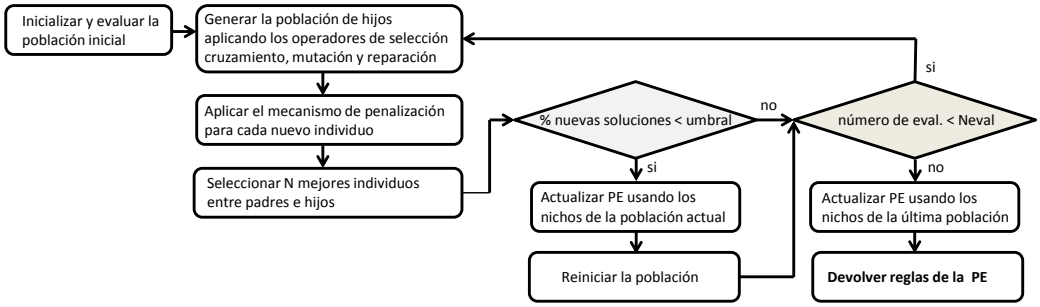

Figura 4.3: Organigrama del método NIGAR

4.1.7. Pasos del algoritmo

De acuerdo con la descripción anterior, el algoritmo propuesto para extraer RACPNs se puede resumir en los siguientes pasos:

Entrada: tamaño de la población N, número de evaluaciones N_{eval}, probabilidad de mutación P_{mut}, factor de amplitud para cada atributo de la BD δ, umbral de diferencia α, umbral de formación de nichos $Nich_{Min}$, umbral de calidad Ev_{Min}.

Salida: Reglas de la PE

Paso 1: Inicialización

1.1 Generar la población inicial (P_t) con N cromosomas.

1.2 Evaluar la población inicial.

Paso 2: Generar la población de hijos (Q_t) de la siguiente manera:

2.1 Seleccionar un par de padres utilizando selección por torneo binario basada en la evaluación en la función objetivo de los cromosomas.

2.2 Se aplica el operador de cruce sobre el par de padres seleccionado, este operador intercambia los genes de los dos padres y aplica el operador PCBLX sobre los límites de los intervalos de los atributos numéricos. Después, se aplica el operador de mutación y reparación sobre los dos hijos generados.

2.3 Evaluar los nuevos individuos.

2.4 Si la PE no está vacía, se aplica el mecanismo de penalización sobre los nuevos individuos si pertenecen a algún nicho de la PE y su evaluación es menor que la evaluación del mejor individuo de su nicho.

2.5 Si la población de hijos no se ha completado, ir al paso a).

Paso 3: Seleccionar los N mejores individuos entre padres e hijos para construir la siguiente población (P_{t+1}).

Paso 4: Si la diferencia entre la población actual y la población anterior es menor que α %:

4.1 Actualizar la PE con la mejor regla de cada nicho de P_{t+1}.

4.2 Reiniciar la población y aplicar el mecanismo de penalización sobre los nuevos individuos que han sido creados en este proceso.

Paso 5: Si no se ha alcanzado el número máximo de evaluaciones, ir al Paso 2.

Paso 6: Actualizar la PE con la mejor regla de cada nicho de la última población.

Paso 7: Devolver las reglas de la PE.

4.2. Estudio Experimental

En esta sección hemos desarrollado varios experimentos para evaluar la efectividad de nuestra propuesta sobre 26 BDs del mundo real. La organización de esta sección es la siguiente:

- La subsección 4.2.1 presenta las BDs usadas en estos experimentos y la configuración de los parámetros de los algoritmos que utilizaremos en este estudio.
- La subsección 4.2.2 analiza la influencia de los umbrales $Nich_{Min}$ y Ev_{Min} en el rendimiento de nuestra propuesta.
- La subsección 4.2.3 compara el rendimiento de nuestro enfoque con cuatro algoritmos evolutivos (EARMGA [YZZ09], GAR [MAR02], GENAR [MAR01] y Alatasetal [AA06]) para extraer RACPNs.
- La subsección 4.2.4 compara nuestro enfoque con dos algoritmos clásicos para extraer reglas de asociación positivas (Apriori [SA96] y Eclat [Zak00]) y con otro AGN clásico (Clearing [Pét96, Pét97]).
- La subsección 4.2.5 analiza la escalabilidad de nuestro enfoque.
- La subsección 4.2.6 presenta un estudio sobre la diversidad del conjunto de reglas obtenido por nuestra propuesta.

4.2.1. Experimentos

En este estudio experimental hemos utilizado las mismas 26 BDs utilizadas en el capítulo 2 (ver subsección 2.2.1 del capítulo 2), las cuales pueden ser descargadas del repositorio de datos KEEL-dataset [AFFL$^+$11]. Para desarrollar los diferentes experimentos, consideramos los resultados promedio de 5 ejecuciones para cada BD.

En este estudio hemos comparado el rendimiento de nuestro enfoque con los algoritmos evolutivos Alatasetal [AA06], GENAR [MAR01], EARMGA [YZZ09] y GAR [MAR02] para extraer RACPNs (ver subsección 1.4.1 del capítulo 1). Además, comparamos nuestra propuesta con los algoritmos clásicos Apriori [Bor03, SA96] y Eclat [Zak00] (ver subsección 1.3 del capítulo 1) y con el AGN clásico Clearing [Pét96, Pét97] (ver subsección 1.4.3 del capítulo 1).

Para comparar nuestra propuesta con el algoritmo Clearing, hemos extendido este método para extraer RACPNs utilizando el mismo esquema de codificación, función objetivo, población inicial y operadores genéticos que en nuestra propuesta (ver las subsecciones de la 4.1.2 a la 4.1.4). Además, utilizamos la misma medida de distancia que nuestro enfoque, teniendo en cuenta los dos componentes de las reglas para determinar si dos individuos pertenecen a un mismo nicho (ver subsección 4.1.1).

Los parámetros utilizados para los métodos analizados se muestran en la Tabla 4.2. Para nuestra propuesta hemos seleccionado valores que funcionan bien en la mayoría de las BDs en lugar de buscar valores específicos para cada BD. Para el resto de los algoritmos hemos seleccionado valores para los parámetros de acuerdo a las recomendaciones de los autores de cada propuesta. Como comentamos en la subsección 2.2.1 del capítulo 2 Apriori, Eclat y GAR necesitan un mínimo de soporte y mínimo de confianza para extraer RACs. Para facilitar las comparaciones hemos seleccionando los valores 0,1 y 0,8 para el mínimo de soporte y confianza, respectivamente, los cuales representan valores estándar que funcionan bien en la mayoría de los casos para todas las BDs.

4.2.2. Análisis de la influencia de algunos parámetros sobre NIGAR

En esta sección, hemos realizado diferentes experimentos para analizar la influencia de los umbrales $Nich_{Min}$ y Ev_{Min} en el rendimiento de nuestra propuesta. Para facilitar la interpretación de este análisis hemos usado tres valores

Tabla 4.2: Parámetros considerados en la comparación de los métodos

Algorithms	*Parameters*
Apriori	*mínimo de soporte = 0,1, mínimo de confianza = 0,8*
Eclat	*mínimo de soporte = 0,1, mínimo de confianza = 0,8*
Alatasetal	N_{eval}=100000, nCromoInicialAleat=12, $r = 3$, $TamTorneo = 10$, P_{sel}= 0,25, $P_{cru} = 0,7$, $P_{mut\_min} = 0,05$, $P_{mut\_max} = 0,9$, $Peso_{sop} = 5$, $Peso_{conf} = 20$, $Peso_{amplRule} = 0,05$, $Peso_{amplInterv} = 0,02$, $Peso_{cubrimiento} = 0,01$
EARMGA	$TamPop = 100$, $N_{eval} = 100000$, $k = 2$, $P_{sel} = 0,75$, $P_{cru} = 0,7$, $P_{mut} = 0,1$, $\alpha = 0,01$
GENAR	$TamPop = 100$, $N_{eval} = 100000$, P_{sel}= 0,25, $P_{cru} = 0,7$, $P_{mut} = 0.1$, $nReglas = 30$, $FP = 0,7$, $AF = 0,2$
GAR	$TamPop = 100$, $nItems = 100$, $N_{eval} = 100000$, P_{sel}= 0,25, $P_{cru} = 0,7$, $P_{mut} = 0,1$, $\omega = 0,4$, Ψ= 0,7, μ= 0,5, *mínimo de soporte = 0,1, mínimo de confianza = 0,8*
Clearing	$TamPop = 100$, N_{eval}=100000, P_{mut}= 0,1, γ=3, $Nich_{Min} = 0,5$
NIGAR	$TamPop = 100$, N_{eval}=100000, P_{mut}= 0,1, γ=3, $Nich_{Min} = 0,5$, Ev_{Min}= 85 %, α = 5 %

Tabla 4.3: Análisis del rendimiento de nuestra propuesta dependiendo del umbral $Nich_{Min}$ con el umbral $Ev_{Min} = 85\,\%$

$Nich_{Min}$	$\#R$	Med_{Sop}	Med_{Conf}	Med_{Lift}	Med_{Conv}	Med_{FC}	$Med_{Netconf}$	$Med_{Yule'sQ}$	Med_{Amp}	Med_{Div}	$\%Ejem$
NIGAR-σ-0,4	18,68	0,21	0,93	8,87	∞	**0,88**	**0,83**	**0,98**	**2,08**	**0,86**	92,63
NIGAR-σ-0,5	20,06	0,22	**0,93**	**8,93**	∞	**0,88**	**0,83**	0,97	2,09	0,85	95,4
NIGAR-σ-0,6	21,15	**0,22**	0,92	7,67	∞	0,87	0,82	0,97	**2,08**	0,83	**95,7**

diferentes para el umbral $Nich_{Min}$ (0‘4, 0‘5 y 0‘6) y el umbral Ev_{Min} (80 %, 85 % y 90 %). Los resultados medios obtenidos se muestran en las Tablas 4.3 y 4.4, donde $\#R$ representa el número medio de reglas, Med_{Sop}, Med_{Conf}, Med_{Lift}, Med_{Conv}, Med_{FC}, $Med_{Netconf}$, Med_{YulesQ} representan los valores medios de soporte, confianza, lift, conviction, FC, netconf y yule'sQ, respectivamente, Av_{Amp} es el número medio de atributos involucrados en las reglas, Med_{Div} representa el valor medio de diversidad de las reglas obtenidas y $\%Reg$ es el tanto por ciento de registros de la BD cubiertos por las reglas generadas. La diversidad de un conjunto de reglas es igual a la media de distancia acir (ver ecuación 4.4) de cada regla al resto de las reglas.

Tabla 4.4: Análisis del rendimiento de nuestra propuesta dependiendo del umbral Ev_{Min} con el umbral $Nich_{Min} = 0{,}5$

$Nich_{Min}$	$\#R$	Med_{Sop}	Med_{Conf}	Med_{Lift}	Med_{Conv}	Med_{FC}	$Med_{Netconf}$	$Med_{Yule'sQ}$	Med_{Amp}	Med_{Div}	$\%Ejem$
NIGAR-λ-80	22,44	0,21	**0,93**	**9,12**	∞	0,86	0,82	0,96	2,10	**0,85**	**96,43**
NIGAR-λ-85	20,06	0,22	**0,93**	8,93	∞	0,88	0,83	0,97	2,09	**0,85**	95,4
NIGAR-λ-90	**16,24**	**0,22**	**0,93**	8,07	∞	**0,89**	**0,84**	**0,98**	**2,08**	0,84	89,24

Analizando los resultados presentados en la Tabla 4.3 podemos ver como disminuye el número de reglas cuando disminuye el valor del umbral $Nich_{Min}$. Este efecto ocurre porque es más fácil superar el umbral para que las reglas pertenezcan a un mismo nicho, agrupándose más reglas en cada nicho. Esto provoca una reducción en el número de nichos generados y, por lo tanto, en el número de reglas que obtenemos. De esta manera, obtenemos un conocimiento muy distinto sobre la BD porque las reglas tendrán que ser bien distintas para pertenecer a nichos diferentes. Sin embargo, esto implica que no puedan crearse nichos más específicos en la población, aumentando las zonas del espacio de búsqueda de donde difícilmente se podrá extraer información, lo que disminuirá el cubrimiento de la BD.

La Tabla 4.4 muestra como aumentan los valores de las medidas de interés cuando aumenta el valor del umbral Ev_{Min}, porque exigimos una mayor calidad de las reglas para que puedan entrar en la PE. Sin embargo, también se puede observar como disminuye el cubrimiento de la BD al aumentar el valor de este umbral, pues resulta muy difícil extraer conocimientos de muy elevada calidad de todo el espacio de búsqueda.

Nosotros, para el resto de los experimentos utilizaremos 0,5 para el umbral $Nich_{Min}$ y 85 % para el umbral Ev_{Min} ya que estos valores nos permiten mantener un buen equilibrio entre el número de reglas, la calidad y el cubrimiento de la BD.

4.2.3. Comparación con otros métodos evolutivos

Esta sección compara el rendimiento de nuestra propuesta con cuatro algoritmos evolutivos (EARMGA [YZZ09], GAR [MAR02], GENAR [MAR01] y Alatasetal [AA06]) para extraer RACPNs. La Tabla 4.5 muestra los resultados medios obtenidos por los métodos en todas las BD (la cabecera de esta tabla ha sido

Tabla 4.5: Resultados del valor medio de las medidas para todas las BDs en la comparación entre NIGAR y otros enfoques evolutivos

Algoritmos	*#R*	Med_{Sop}	Med_{Conf}	Med_{Lift}	Med_{Conv}	Med_{FC}	$Med_{Netconf}$	$Med_{Yule'sQ}$	Med_{Amp}	Med_{Div}	*%Reg*
EARMGA	**95,3**	0,35	**1**	1,02	∞	0,05	0,01	0,02	**2,01**	0,58	**99,28**
GAR	54	**0,6**	0,91	1,47	∞	0,42	0,34	0,67	2,11	0,23	87,06
GENAR	29,5	0,24	0,83	3,33	∞	0,58	0,38	0,67	28,88	0,32	66,64
Alatasetal	23,55	0,44	0,99	**18,15**	∞	0,49	0,21	0,38	5,64	0,22	77,44
NIGAR	20,06	0,22	0,93	8,93	∞	**0,88**	**0,83**	**0,97**	2,09	**0,85**	95,4

introducida en la sección anterior). Los resultados obtenidos por cada método en cada BD se pueden encontrar en la Tablas C.14, C.15, C.16 y C.17 de la sección C.3 del apéndice C.

Estos resultados muestran como nuestra propuesta obtiene un conjunto reducido de reglas cortas (el método que menos reglas obtiene con solo 20 en media), evitando obtener reglas redundantes que proporcionen información similar sobre la BD. Además, estas reglas ofrecen un conocimiento sobre toda la BD, obteniendo un alto cubrimiento en todas las BDs. Destacar, que solo EARMGA obtiene mejores valores de cubrimiento que nuestra propuesta, pero sus reglas presentan los peores valores para las medidas de interés.

Para analizar los resultados obtenidos por las medidas de calidad lift, FC, netconf y yule'sQ (aplicando las transformaciones a los valores medios de estas medidas descritas en la subsección 2.2.2 del capítulo 2), y la medida de diversidad del conjunto de reglas hemos utilizado tests no paramétricos para comparaciones múltiples.

En primer lugar, hemos utilizado el test de Friedman [Fri37] y el test de Iman-Davenport [ID80] para determinar si existen diferencias significativas entre todos los valores medios. Los resultados de estos tests se muestran en la Tabla 4.6, donde podemos ver que se rechaza la hipótesis de igualdad obteniendo p-valores inferiores a 0,0001 con un nivel de confianza $\alpha = 0{,}05$. La Tabla 4.7 muestra los rankings (calculados utilizando el test de Friedman) de los algoritmos analizados, donde se aprecia que nuestra propuesta obtuvo el mejor ranking para todas las medidas.

Después de confirmar que existen diferencias significativas entre los algoritmos, se aplica el test de Holm [Hol79] y el test de Finner [Fin93] para comparar el algoritmo de mejor ranking (NIGAR) con el resto de los algoritmos. La Tabla 4.8

Tabla 4.6: Resultados del test de Friedman e Iman-Davenport ($\alpha = 0{,}05$) en la comparación entre los algoritmos evolutivos y NIGAR

Test de Friedman					
	FC	**Netconf**	**Yule'sQ**	**Lift**	**Diversidad**
Valor Crítico	11,07	11,07	11,07	11,07	11,07
Estadística (X_F^2)	**54,36**	**76,51**	**78,24**	**50,96**	**79**
p **valor**	<0,0001	<0,0001	<0,0001	<0,0001	<0,0001
Test de Iman-Davenport					
	FC	**Netconf**	**Yule'sQ**	**Lift**	**Diversidad**
Valor Crítico	2,29	2,29	2,29	2,29	2,29
Estadística (F_F)	**27,37**	**69,59**	**75,95**	**24,02**	**79,03**
p **valor**	<0,0001	<0,0001	<0,0001	<0,0001	<0,0001

Tabla 4.7: Ranking promedio de los algoritmos evolutivos en la comparación con NIGAR

Algoritmos	**Rankings**				
	FC	**Netconf**	**Yule'sQ**	**Lift**	**Diversidad**
EARMGA	4,40	4,53	4,57	4,46	2,23
Alatasetal	3,69	4,03	4,13	3,48	4,36
GAR	3,01	2,67	2,78	3,13	3,98
GENAR	2,46	2,67	2,28	2,38	3,42
NIGAR	**1,42**	**1,07**	**1,21**	**1,53**	**1**

presenta estos resultados, donde los métodos son ordenados respecto al z-valor obtenido. El test de Holm y el test de Finner rechazan la hipótesis de igualdad con el resto de los algoritmos ($p < \alpha/i$) en todas las medidas con un nivel de confianza $\alpha = 0{,}05$. Por lo tanto, podemos concluir que NIGAR obtiene el mejor rendimiento para todas las medidas en comparación con el resto de los métodos usados en este estudio. Destacar que aunque los test estadísticos demuestran que NIGAR es mejor que Alatasetal con diferencias significativas, este método obtuvo mejor valor medio que nuestra propuesta para la medida lift porque alcanzó valores muy altos en algunas BDs, debido a que esta medida no tiene límite superior (ver subsección 1.2.1 del capítulo 1).

Tabla 4.8: Resultados de los tests de Holm y Finner ($\alpha = 0{,}05$) en la comparación entre los algoritmos evolutivos y NIGAR

i	Algoritmos	z	p	Holm	Finner	**Hipótesis**
			FC			
4	EARMGA	6,79	0	0,012	0,012	**Rechazada**
3	Alatasetal	5,17	0	0,016	0,025	**Rechazada**
2	GAR	3,64	0,000273	0,025	0,037	**Rechazada**
1	GENAR	2,36	0,017882	0,05	0,05	**Rechazada**
			Netconf			
4	EARMGA	7,89	0	0,012	0,012	**Rechazada**
3	Alatasetal	6,75	0	0,016	0,025	**Rechazada**
2	GAR	3,64	0,000273	0,025	0,037	**Rechazada**
1	GENAR	3,63	0,000273	0,05	0,05	**Rechazada**
			Yule'sQ			
4	EARMGA	7,67	0	0,012	0,012	**Rechazada**
3	Alatasetal	6,66	0	0,016	0,025	**Rechazada**
2	GAR	3,59	0,000323	0,025	0,037	**Rechazada**
1	GENAR	2,45	0,014059	0,05	0,05	**Rechazada**
			Lift			
4	EARMGA	6,67	0	0,012	0,012	**Rechazada**
3	Alatasetal	4,42	0,000009	0,016	0,025	**Rechazada**
2	GAR	3,63	0,000273	0,025	0,037	**Rechazada**
1	GENAR	1,92	0,053665	0,05	0,05	**Rechazada**
			Diversidad			
4	Alatasetal	7,67	0	0,012	0,012	**Rechazada**
3	GAR	6,79	0	0,016	0,025	**Rechazada**
2	GENAR	5,52	0	0,025	0,037	**Rechazada**
1	EARMGA	2,80	0,005007	0,05	0,05	**Rechazada**

Las Figuras 4.4 y 4.5 representan boxplots que muestran los valores de las medidas FC y netconf, respectivamente, para las reglas obtenidas en una de las 5 ejecuciones realizadas por NIGAR para todas las BDs. Podemos ver como todas las reglas representan dependencias positivas y más del 75 % de las reglas obtenidas tienen un valor superior a 0.7 para el FC y 0.6 para netconf, menos para la BD stulong.

Las Figuras 4.6 y 4.7 son boxplots que muestran los valores para las medidas FC y netconf de las reglas obtenidas a partir de una de las 5 ejecuciones realizadas por todos los algoritmos evolutivos analizados en la BD stock. Observe que NIGAR presenta mejores valores de FC y netconf que el resto de los algoritmos comparados, obteniendo la mayoría de sus reglas valores cercanos al mejor valor posible para estas medidas. Además, podemos ver cómo EARMGA y Alatasetal obtienen algunas reglas que representan independencia o dependencia negativa entre sus ítems según estas medidas.

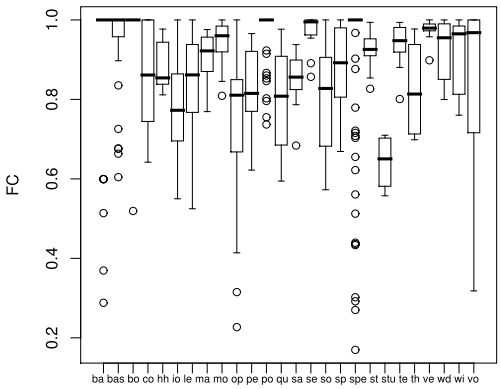

Figura 4.4: Boxplot de la medida FC para NIGAR en todas las BDs

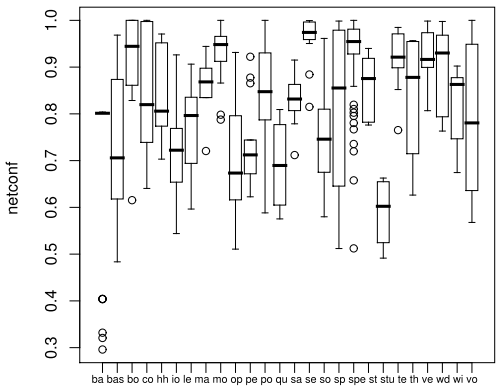

Figura 4.5: Boxplot de la medida netconf para NIGAR en todas las BDs

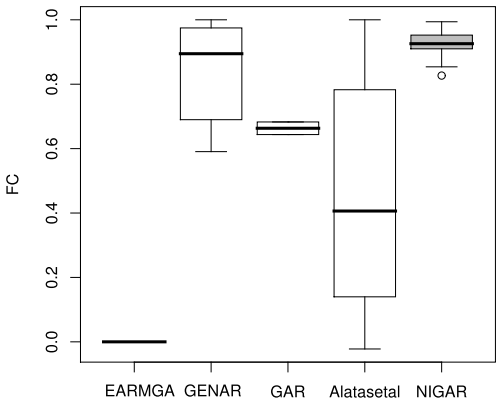

Figura 4.6: Boxplot de la medida FC para los métodos evolutivos y NIGAR para la BD stock

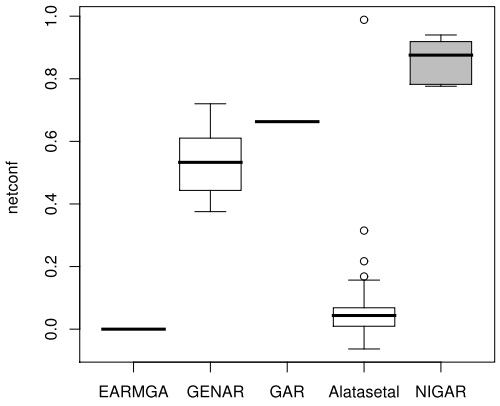

Figura 4.7: Boxplot de la medida netconf para los métodos evolutivos y NIGAR para la BD stock

4.2.4. Comparación con los algoritmos clásicos

En esta sección realizamos un análisis comparativos con dos algoritmos clásicos Apriori [SA96] y Eclat [Zak00] y con el AGN Clearing [Pét96, Pét97]. Para comparar NIGAR con los métodos clásicos Apriori y Eclat, al igual que en la sección 2.2.4 del capítulo 2, hemos utilizado un particionamiento en amplitud en cada atributo cuantitativo [LHTD02].

Como hemos comentado anteriormente, para comparar el AGN clásico Clearing con nuestra propuesta, hemos extendido este método para extraer reglas de asociación, utilizando el mismo esquema de codificación, función objetivo, población inicial, operadores genéticos y medida de distancia entre reglas (ver subsección 4.1.1) que en nuestra propuesta. Los resultados medios obtenidos por los métodos en todas las BD se muestran en la Tabla 4.9 (la cabecera de esta tabla ha sido introducida en la sección 4.2.2). Los resultados obtenidos por cada método en cada BD se pueden encontrar en las Tablas C.18 y C.19 de la sección C.3 del apéndice C. Para Apriori y Eclat solo se muestran los resultados obtenidos sobre 15 BDs, por los problemas de escalabilidad que ellos presentan. Para analizar los resultados obtenidos por las medidas de calidad lift, FC, netconf y yule'sQ, y la medida de diversidad del conjunto de reglas hemos utilizado el test de Wilcoxon [She03, Wil45] con un nivel de confianza $alpha = 0{,}05$. La Tabla 4.10 muestra los resultados del test de Wilcoxon.

Como podemos ver en la Tabla 4.9 NIGAR obtiene el conjunto de reglas más reducido y con el mejor valor medio de cubrimiento de las BDs. Si nos centramos en los resultados obtenidos con los test estadísticos podemos observar como la hipótesis de igualdad es rechazada en todas las medidas de interés excepto en lift, debido a que esta medida no tiene límite superior y estos métodos obtienen valores muy altos para algunas BDs, además para los algoritmos clásicos Apriori y Eclat solo se pudieron analizar los resultados de las 15 BDs donde pudieron ejecutarse.

Tabla 4.9: Resultados del valor medio de las medidas para todas las BDs en la comparación entre NIGAR y los métodos clásicos

Algoritmos	*#R*	Med_{Sop}	Med_{Conf}	Med_{Lift}	Med_{Conv}	Med_{FC}	$Med_{Netconf}$	$Med_{Yule'sQ}$	Med_{Amp}	Med_{Div}	*%Reg*
Apriori	**8345221,46**	0,16	**0,93**	4,09	∞	0,81	0,64	0,84	5,13	0,51	90,43
Eclat	**8345221,46**	0,16	**0,93**	4,09	∞	0,81	0,64	0,84	5,13	0,51	90,43
Clearing	26,08	0,16	0,9	**26,05**	∞	0,83	0,79	0,9	2,1	0,83	92,72
NIGAR	20,06	**0,22**	**0,93**	8,93	∞	**0,88**	**0,83**	**0,97**	**2,09**	**0,85**	**95,4**

Tabla 4.10: Resultados del test de Wilcoxon ($\alpha = 0.05$) en la comparación entre los algoritmos clásicos y NIGAR

Comparación	R^+	R^-	**Hipótesis**	p-**valor**
		FC		
NIGAR vs. Clearing	266,5	58,5	**Rechazada**	0,003
NIGAR vs. Apriori	86	19	**Rechazada**	0,003
NIGAR vs. Eclat	86	19	**Rechazada**	0,003
		Netconf		
NIGAR vs. Clearing	234,5	90,5	**Rechazada**	0,05
NIGAR vs. Apriori	100,5	19,5	**Rechazada**	0,019
NIGAR vs. Eclat	100,5	19,5	**Rechazada**	0,019
		Yule'sQ		
NIGAR vs. Clearing	315,5	35,5	**Rechazada**	<0,001
NIGAR vs. Apriori	95	25	**Rechazada**	0,04
NIGAR vs. Eclat	95	25	**Rechazada**	0,04
		Lift		
NIGAR vs. Clearing	37	314	**No Rechazada**	$\geq$ 0,2
NIGAR vs. Apriori	74	46	**No Rechazada**	$\geq$ 0,2
NIGAR vs. Eclat	74	46	**No Rechazada**	$\geq$ 0,2
		Diversidad		
NIGAR vs. Clearing	218	133	**No Rechazada**	$\geq$ 0,2
NIGAR vs. Apriori	117	3	**Rechazada**	<0,001
NIGAR vs. Eclat	117	3	**Rechazada**	<0,001

Al analizar la medida de diversidad se obtienen diferencias significativas con los métodos Apriori y Eclat, pero no con el método Clearing. Esto se debe a que este AGN fomenta la diversidad en el proceso de búsqueda y hemos utilizado para ello la misma medida de distancia que hemos propuesto para nuestro método para determinar los nichos. Aun así, nuestra propuesta consigue mejor ranking para esta medida de diversidad y para las medidas de interés obtiene diferencias significativas.

4.2.5. Análisis de escalabilidad

En esta sección se presentan varios experimentos para analizar la escalabilidad de los algoritmos en la BD House_16H. Todos los experimentos se realizaron en un procesador Intel Core i7, 2,80 GHz CPU con 12 Gb de memoria y fueron ejecutados en Linux. El tiempo de ejecución promedio empleado por los algoritmos analizados cuando aumentan el número de atributos y ejemplos se muestran en la Tabla 4.11 y Tabla 4.12, respectivamente.

Tabla 4.11: El tiempo de ejecución (segundos) empleado por todos los algoritmos comparados y NIGAR cuando el número de atributos aumenta en la BD House_16H

	Número de Atributos				
Algoritmos	4	8	12	16	17
EARMGA	213	194	216	189	221
GAR	1187	2787	4046	4318	4583
GENAR	58	72	**93**	112	**121**
Alatasetal	99	154	97	135	302
Apriori	**3**	**5**	233	5192	11268
Eclat	**3**	**5**	251	5812	12467
Clearing	69	101	110	**104**	132
NIGAR	89	139	114	131	162

Tabla 4.12: El tiempo de ejecución (segundos) empleado por todos los algoritmos comparados y NIGAR cuando el número de ejemplos aumenta en la BD House_16H

	Número de Ejemplos				
Algoritmos	20 %	40 %	60 %	80 %	100 %
EARMGA	46	89	124	171	221
GAR	911	1720	2407	3055	4583
GENAR	**23**	**47**	**73**	**90**	**121**
Alatasetal	33	48	197	215	302
Apriori	2689	5180	10050	9004	11268
Eclat	3076	8700	11300	10604	12467
Clearing	25	58	74	114	132
NIGAR	32	60	85	108	162

Las Figuras 4.8 y 4.9 muestran la relación entre el tiempo de ejecución y el número de atributos para los algoritmos evolutivos y los algoritmos clásicos estudiados, respectivamente. Además, las Figuras 4.10 y 4.11 muestran la relación entre el tiempo de ejecución y el número de ejemplos para dichos algoritmos. Podemos ver cómo la mayoría de los métodos tienden a aumentar linealmente el

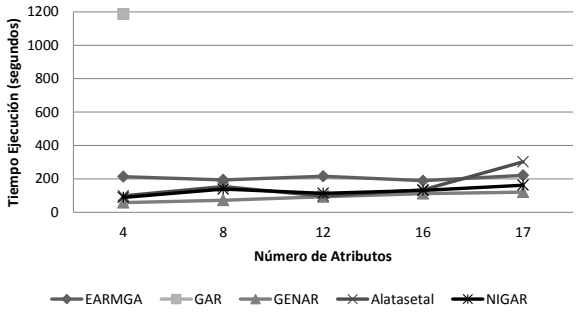

Figura 4.8: Relación entre el tiempo de ejecución y el número de atributos para los algoritmos evolutivos y NIGAR en la BD House_16H

tiempo de ejecución cuando aumenta el número de atributos y ejemplos de la BD, exceptuando a Apriori, Eclat y GAR, los cuales aumentan de manera exponencial. Las Figuras 4.8 y 4.10 muestran pocos resultados sobre GAR, porque su tiempo de ejecución es superior a más de 1200 segundos en casi todos los casos. Como hemos comentado, los tiempos de ejecución de GAR son más altos que el resto de los algoritmos evolutivos porque necesita un proceso adicional para extraer las reglas de asociación.

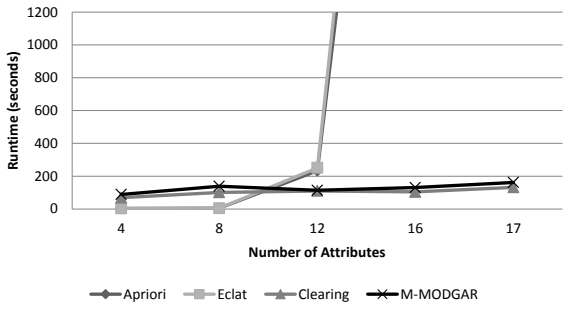

Figura 4.9: Relación entre el tiempo de ejecución y el número de atributos para los algoritmos clásicos y NIGAR en la BD House_16H

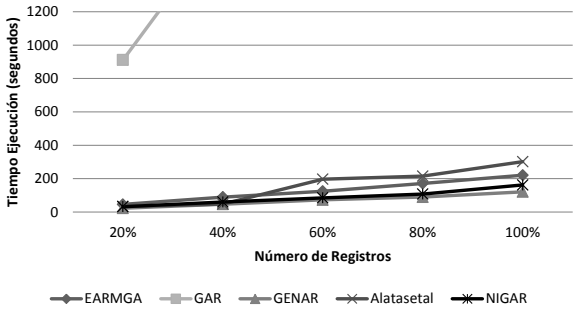

Figura 4.10: Relación entre el tiempo de ejecución y el número de ejemplos para los algoritmos evolutivos y NIGAR en la BD House_16H

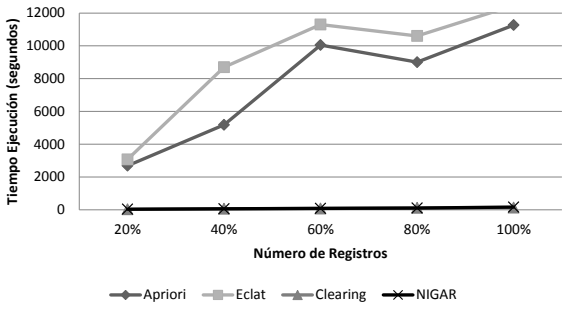

Figura 4.11: Relación entre el tiempo de ejecución y el número de ejemplos para los algoritmos clásicos y MOPNAR en la BD House_16H

4.2.6. Análisis de la diversidad de conjuntos de reglas obtenidos por algunos métodos evolutivos

En esta sección analizamos la diversidad de las reglas obtenidas por nuestra propuesta y el resto de los algoritmos estudiados. La Figura 4.12 representa un boxplot que muestra los valores correspondientes a la medida de diversidad (ver subsección 4.2.2) para las reglas obtenidas en una de las 5 ejecuciones realizadas por nuestra propuesta para todas las BDs. Como podemos ver todos los conjuntos

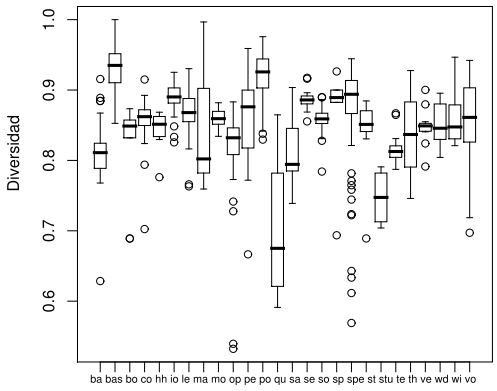

Figura 4.12: Boxplot de la medida diversidad para NIGAR en todas las BDs

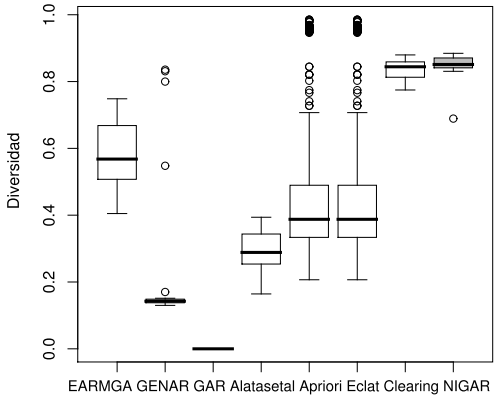

Figura 4.13: Boxplot de la medida diversidad para todos los algoritmos en la BD stock

de reglas presentan una alta diversidad, donde más del 75 % de las reglas obtenidas alcanzan un valor mayor que 0.7 para la medida de diversidad (menos para la BD quake). La Figura 4.13 muestra un boxplot con los valores de la medida de diversidad para las reglas obtenidas por los métodos analizados en la BD stock, seleccionada al azar. Observe cómo los métodos basados en nichos proporcionan un conocimiento más diverso que el resto de los métodos. Además, Apriori y Eclat obtienen un amplio rango de valores de diversidad (desde 0.2 hasta 1), debido a la gran cantidad de reglas que generan para todas las BD.

La Tabla 4.13 muestra algunas de las reglas obtenidas por nuestra propuesta en una de las 5 ejecuciones realizadas en la BD stock, donde $ConjReglas$ representa la regla generada, Sop, FC y $Netconf$ representan el valor de las medidas soporte, FC y netconf de las reglas, respectivamente, y Div es el valor de la medida diversidad de la regla. Podemos ver que todas las reglas ofrecen diferente información sobre el problema y obtienen valores cercanos al mejor valor posible que las medidas de calidad pueden alcanzar. Esto nos permite evitar la generación de reglas que sean redundantes y que dificultan al usuario comprender el conocimiento ofrecido por el método. Por ejemplo la Tabla 4.13 muestra algunas de las reglas obtenidas por los algoritmos Alatasetal y GAR. En estas reglas podemos encontrar varios problemas:

- El método Alatasetal obtiene muchas reglas que son especializaciones de reglas generales con niveles de calidad muy similares, por ejemplo la regla

Tabla 4.13: Reglas obtenidas por algunos métodos evolutivos en la BD stock

Algoritmo	*ConjReglas*	*Sop*	*FC*	*Netconf*	*Div*
NIGAR	R1: If **Company4** is not [47,37, 59,87] then **Company7** is not [74, 85,87]	0.27	0.95	0.93	0.87
	R2: If **Company1** is not [31,89, 61,5] then **Company2** is [49,0, 55,75]	0.33	0.97	0.93	0.84
	R3: If **Company2** is [22,12, 36] then **Company5** is [30,12551,87]	0.24	0.99	0.91	0.86
Alatasetal	R1: If **Company5** is [90,37, 93] then **Company1** is not [31,89, 61,5]	0.02	1	0.66	0
	R2: If **Company5** is [93, 93] then **Company1** is not [31,89, 61,5]	0.001	1	0.65	0
	R3: If **Company5** is [90,375, 93,0] and **Company4** is [44,37, 45,87] then **Company1** is not [31,89, 61,5]	0.02	1	0.66	0
GAR	R1: If **Company3** is [19,24, 22,37] then **Company2** is [49,34, 59,14]	0.39	0.69	0.67	0
	R2: If **Company2** is [49,34, 59,14] then **Company3** is [19,24, 22,37]	0.39	0.64	0.66	0

específica $R2$ y la regla general $R1$. En este caso, solo se debería proporcionar la regla más general, reduciendo el número de reglas proporcionadas al usuario.

- En las reglas $R1$ y $R3$ obtenidas por Alatasetal podemos ver que la adición de más atributos en el antecedente no siempre afecta a la predicción del consecuente. Reglas como $R3$ no son necesarias puesto que son mas difíciles de entender y por lo tanto menos útiles en principio para el usuario.
- Las reglas $R1$ y $R2$ obtenidas por GAR representan la doble implicación de la misma regla. Una opción es proporcionar las dos reglas y que el usuario decida cuál es más interesante. Por otro lado, parece coherente eliminar la regla que presente menor calidad y proporcionar al usuario solo la mejor, reduciendo la complejidad del conjunto de reglas obtenido.
- Otro problema que se consigue evitar es la generación de reglas con los mismos atributos y con un alto solapamiento de los intervalos.

4.3. Sumario

En este capítulo, hemos presentado NIGAR, un nuevo AGN para extraer un conjunto diverso de RACPNs. Este método realiza un aprendizaje evolutivo de los intervalos de los atributos y una selección de las condiciones para cada regla, maximizando una función objetivo que nos permite obtener reglas interesantes, fáciles de comprender y con un buen cubrimiento de la BD. Además, NIGAR utiliza una PE y un mecanismo de penalización para mantener múltiples soluciones en la población y presenta dos umbrales que permiten controlar la calidad y la diversidad de las reglas obtenidas. Finalmente, definimos una nueva medida de distancia entre reglas para determinar los nichos en la población, basada en los atributos comunes y los ejemplos cubiertos de la BD. Del estudio experimental realizado podemos concluir:

- La PE, el mecanismo de penalización y el proceso de reinicialización permiten localizar y mantener múltiples soluciones óptimas, permitiéndonos obtener conjuntos de reglas con un buen equilibrio entre calidad y diversidad del conocimiento obtenido sobre toda la BD y un alto cubrimiento de las BDs.
- Las reglas obtenidas involucran pocos atributos, facilitando su interpretabilidad y comprensión desde el punto de vista del usuario.
- El método propuesto emplea una cantidad razonable de tiempo en todas las BDs y presenta una buena escalabilidad cuando el tamaño del problema aumenta.

Comentarios Finales

Dedicaremos esta sección a la presentación de un resumen de los resultados obtenidos y conclusiones que esta memoria puede aportar. Presentaremos las publicaciones asociadas a esta tesis y comentaremos algunos aspectos sobre trabajos futuros que siguen la línea aquí expuesta y sobre otras líneas de investigación que se pueden derivar.

A. Resumen y Conclusiones

En esta memoria hemos presentado diferentes algoritmos evolutivos para obtener reglas de asociación con una alta calidad y diversidad en el conjunto de reglas obtenido.

En primer lugar, se ha presentado QAR-CIP-NSGA-II, un nuevo modelo evolutivo multi-objetivo para extraer reglas de asociación con un buen equilibrio entre las diferentes medidas de interés y el cubrimiento de la BD. Este modelo extiende el AEMO NSGA-II para realizar un aprendizaje evolutivo de los intervalos de los atributos y una selección de condiciones para cada regla, maximizando tres objetivos: interés, comprensibilidad y rendimiento. Esta propuesta introduce dos nuevos componentes al proceso evolutivo: una PE y un proceso de reinicialización, para almacenar todas las reglas no dominadas encontradas y promover diversidad en la población.

Los resultados obtenidos confirman que los dos nuevos componentes (PE y el proceso de reinicialización) mejoran el cubrimiento de la BD y permiten obtener un número mayor de reglas que el modelo evolutivo clásico, debido a que el número de reglas no está limitado por el tamaño de la población actual. Además, esta propuesta obtiene un conjunto de reglas con valores muy altos para las medidas

de interés, proporcionando al usuario reglas de muy alta calidad.

Luego, se ha presentado MOPNAR, un nuevo modelo evolutivo basado en el AEMO MOEA/D-DE para obtener conjuntos más reducidos de reglas a partir de extraer reglas de asociación positivas y negativas, que nos aporten información útil sobre toda la BD. Este nuevo método realiza un aprendizaje evolutivo de los intervalos de los atributos y una selección de las condiciones para cada regla, introduciendo una PE y un proceso de reinicialización al igual que en el primer método presentado, porque estos componentes han demostrado ser muy útiles para promover diversidad en la población, almacenar todas las reglas no dominadas encontradas y mejorar el cubrimiento de la BD. Además, este método maximiza tres objetivos: interés, comprensibilidad y rendimiento, lo que nos permite obtener reglas que sean fáciles de entender, interesantes y con un buen cubrimiento de la BD.

Del estudio realizado podemos concluir que la extracción de reglas de asociación positivas y negativas permite reducir el número de reglas necesarias para obtener conocimiento interesante de toda la BD. Además, el uso del AEMO MOEA/D-DE en esta propuesta nos permite extraer conjuntos de reglas de asociación con un mejor equilibrio entre el número de reglas, el soporte y el cubrimiento de la BD que el AEMO NSGA-II. MOPNAR obtiene conjuntos reducidos de reglas de asociación con valores altos para las medidas de interés y con pocos atributos, lo que permite una mejor comprensión del usuario.

Por último, se ha presentado NIGAR, un nuevo AGN para extraer un conjunto reducido y diverso de reglas de asociación positivas y negativas de alta calidad, comprensibles y con un buen cubrimiento de la BD. Este método realiza un aprendizaje evolutivo de los intervalos de los atributos y una selección de las condiciones para cada regla, utilizando una PE, un mecanismo de penalización y un proceso de reinicialización para localizar y mantener múltiples soluciones óptimas en la población. En esta propuesta se introducen dos umbrales que permiten ajustar el equilibrio entre la diversidad y la calidad de las reglas obtenidas. Además, presentamos una nueva medida de distancia basada en dos componentes de las reglas: el ratio de solapamiento de los atributos comunes y los ejemplos cubiertos por las dos reglas.

Después de realizar diferentes estudios sobre el método, se ha comprobado que permite obtener conjuntos reducidos de reglas con un buen equilibrio entre calidad y diversidad del conocimiento obtenido, evitando la obtención de reglas redundantes que proporcionen información similar sobre la BD. Esta propuesta

obtiene reglas de asociación interesantes, con altos valores para las medidas de interés, un alto cubrimiento de la BD y pocos atributos, facilitando su interpretabilidad desde el punto de vista del usuario.

Destacar que los tres métodos evolutivos permiten obtener reglas de asociación interesantes, por lo que su uso dependerá de las necesidades específicas de cada usuario. Los usuarios podrán basar su selección en las potencialidades que cada método ofrece, donde se destaca la extracción de reglas de asociación más específicas de muy alta calidad por el método QAR-CIP-NSGA-II, la obtención de un conjunto más reducido de reglas de asociación interesantes con el mejor cubrimiento de las BDs por el método MOPNAR y la obtención de un conjunto muy diverso de reglas de asociación con buena calidad y cubrimiento de la BD por el método NIGAR, el cual obtiene reglas más generales que QAR-CIP-NSGA-II.

B. Publicaciones Asociadas a la Tesis

A continuación se presenta un listado de las publicaciones asociadas a la tesis.

- Publicaciones en revistas internacionales:

  1. D. Martín, A. Rosete, J. Alcalá-Fdez, F. Herrera, QAR-CIP-NSGA-II: A New Multi-Objective Evolutionary Algorithm to Mine Quantitative Association Rules, Information Sciences 258 (2014) 1-28. doi: 10.1016/j.ins.2013.09.009
  2. D. Martín, A. Rosete, J. Alcalá-Fdez, F. Herrera, A New Multi-Objective Evolutionary Algorithm for Mining a Reduced Set of Interesting Positive and Negative Quantitative Association Rules. IEEE Transactions on Evolutionary Computation 18:1 (2014) 54-69, doi: 10.1109/TEVC.2013.2285016.
  3. D. Martín, A. Rosete, J. Alcalá-Fdez, F. Herrera, NIGAR: a Niching Genetic Algorithm to Mine a Diverse Set of Interesting Quantitative Association Rules. IEEE Transactions on Knowledge and Data Engineering. (sometido)

- Publicaciones en congresos internacionales:

  1. D. Martín, A. Rosete, J. Alcalá-Fdez, F. Herrera, A Multi-Objective Evolutionary Algorithm for Mining Quantitative Association Rules.

11th International Conference on Intelligent Systems Design and Applications (ISDA 2011), Córdoba (Spain), 1397-1402, 22-24 November 2011.

- Publicaciones en congresos nacionales:

  1. D. Martín, A. Rosete, J. Alcalá-Fdez, F. Herrera, MOPNAR: Algoritmo Evolutivo Multi-Objetivo para Extraer Reglas de Asociación Cuantitativas Positivas y Negativas. Congreso Español sobre Metaheurísticas, Algoritmos Evolutivos y Bioinspirados (MAEB 2013), Madrid (España), pp. 882-891, 17-20 Septiembre, 2013. (*Premio de la AEPIA: mejor artículo de carácter metodológico presentado a MAEB* 2013).

C. Líneas de Investigación Futuras

A continuación, consideraremos algunas líneas de trabajo futuras a partir de las conclusiones obtenidas en esta memoria.

C.1 Extensión de los Métodos Propuestos al Aprendizaje de Reglas de Asociación Difusas

Los métodos presentados en esta memoria han obtenido reglas de asociación de muy alta calidad a partir de datos numéricos. Por eso consideramos, que su adaptación hacia el aprendizaje de reglas de asociación difusas puede obtener interesantes resultados. Las reglas de asociación difusas nos permiten extender los tipos de relaciones que se pueden representar, facilitando la interpretación de las reglas en términos lingüísticos y eludiendo las fronteras no naturales en el particionamiento del dominio de los atributos [DMSaV03, DPS05].

C.2 Adaptación de los Métodos Propuestos a otras técnicas de Minería de Datos

En esta memoria hemos presentado diferentes métodos evolutivos para extraer reglas de asociación. La adaptación de estos métodos pudiera ser muy útil para otras técnicas de minería de datos interesantes, como el descubrimiento de subgrupos [Klo96] y la clasificación asociativa [LHM98, THA07].

El descubrimiento de subgrupos [Klo96] representa una técnica para descubrir reglas interesantes con respecto a una variable objetivo. Esta técnica se encuentra entre las técnicas de extracción de reglas de asociación y la obtención de reglas de clasificación. Esto es debido a que un algoritmo de reglas de asociación obtiene relaciones entre los ítems de la BD, y cualquier atributo puede estar en el antecedente o en el consecuente de la regla, sin embargo en el descubrimiento de subgrupos se prefija el atributo que estará en el consecuente de la regla, el cual representa la variable de interés para el usuario.

Muchos de los algoritmos propuestos [HCGdJ10] para descubrir subgrupos representan adaptaciones de los algoritmos de extracción de reglas de asociación, por las ventajas que ellos ofrecen para esta nueva técnica. El descubrimiento de subgrupos es un campo emergente que tiene muchos problemas abiertos en los que nos podemos enfocar para adaptar nuestros algoritmos al descubrimiento de subgrupos de mejor calidad. Algunos de estos problemas se presentan en [HCGdJ10], destacándose la determinación de cuáles medidas de calidad pueden ser más útiles para evaluar los subgrupos y al mismo tiempo para guiar el proceso de búsqueda, la influencia de la discretización de las variables continuas en la calidad de los subgrupos obtenidos, o las ventajas de los algoritmos de descubrimiento de subgrupos que utilizan las variables continuas sin previa discretización.

Por otra parte, se han presentado varias propuestas [LHM98, THA07] que usan las ventajas de las técnicas de extracción de reglas de asociación en la creación de modelos de clasificación, bajo el nombre de clasificación asociativa. Como se ha dicho, las técnicas de extracción de reglas de asociación tienen como objetivo encontrar relaciones interesantes entre los ítems de la BD, mientras que las técnicas de clasificación tienen como objetivo descubrir un modelo a partir de datos de entrenamiento, para utilizarlo en predecir la clase de los patrones de prueba. Ambas técnicas son esenciales en las aplicaciones reales de minería de datos, por lo que su integración podría resultar de gran interés para los usuarios.

C.3 Diseño de nuevos algoritmos evolutivos para la extracción de reglas de asociación para problemas con características especiales

Los avances alcanzados en el desarrollo de algoritmos evolutivos para la extracción de reglas de asociación nos permiten enfocarnos en realizar nuevos estudios sobre la extracción de reglas de asociación para problemas con características especiales, por ejemplo baja calidad en los datos y grandes volúmenes de datos (*big data*).

En la actualidad los métodos para extraer reglas de asociación se han enfocado en BDs con valores precisos, sin embargo, muchos datos en las aplicaciones del mundo real tienen un cierto grado de imprecisión (por ejemplo un intervalo o valores difusos). El diseño de algoritmos que sean capaces de tratar la incertidumbre de los datos y explotar mejor la información contenida en los conjuntos de datos de baja calidad representa un desafío para los investigadores de este campo [PSC11, VOOS09].

Por otra parte, la generación y almacenamiento de grandes conjuntos de datos ha impulsado aún más el proceso de análisis y extracción de conocimiento, con la creencia de que con más datos disponibles la información resultante podría ser más precisa. Sin embargo, los algoritmos estándar que se utilizan en la minería de datos no suelen ser capaces de tratar con cantidades extremadamente grandes de datos [Sat12]. Por esto, los algoritmos para extraer reglas de asociación deben ser rediseñados y adaptados teniendo en cuenta las soluciones que se utilizan para tratar grandes volúmenes de datos, de manera que puedan utilizarse bajo estas condiciones, manteniendo la calidad del conjunto de reglas obtenido.

Apéndices

Apéndice A

Algoritmos Genéticos

Puesto que todos los métodos propuestos en esta memoria están basados en el uso de los algoritmos genéticos como técnica de optimización y búsqueda, dedicaremos este primer apéndice a describir las líneas generales de este tipo de algoritmos.

Los algoritmos genéticos son algoritmos de búsqueda de propósito general que se basan en principios inspirados en la genética de las poblaciones naturales para llevar a cabo un proceso evolutivo sobre soluciones de problemas. Fueron inicialmente propuestos por Holland [HR77] y han sido posteriormente estudiados en profundidad por otros autores [Gol89, Mic96]. Los algoritmos genéticos han demostrado ser, tanto desde un punto de vista teórico como práctico, una herramienta óptima para proporcionar una búsqueda robusta en espacios complejos, ofreciendo un enfoque válido para solucionar problemas que requieran una búsqueda eficiente y eficaz.

Los algoritmos genéticos se han aplicado con mucho éxito en problemas de búsqueda y optimización. La razón de gran parte de este éxito se debe a su habilidad para explotar la información que van acumulando sobre el espacio de búsqueda que manejan, desconocido inicialmente, lo que les permite redirigir posteriormente la búsqueda hacia subespacios útiles. La *capacidad de adaptación* que presentan es su característica principal, especialmente en espacios de búsqueda grandes, complejos y con poca información disponible, en los que las técnicas clásicas de búsqueda (enumerativas, heurísticas, ...) no presentan buenos resultados.

La idea básica de estos algoritmos consiste en mantener una población de individuos que codifican soluciones del problema. Dichos individuos emplean una representación genética para codificar los valores de las características parciales que definen las distintas soluciones. Debido a ello, cada individuo recibe el nombre de *cromosoma* y cada una de sus componentes el de *gen*.

Los cromosomas se generan inicialmente a partir de la información disponible sobre el problema, o bien de un modo aleatorio cuando no se dispone de esta información, y la población se hace evolucionar a lo largo del tiempo mediante un proceso de competición y alteración controlada que emula los procesos genéticos que tienen lugar en la naturaleza. A lo largo de sucesivas iteraciones, denominadas *generaciones*, los cromosomas se ordenan con respecto a su grado de adaptación al problema, es decir, con respecto a lo bien que resuelven dicho problema y, tomando como base estas evaluaciones, se construye una nueva población mediante un proceso de *selección* y una serie de operadores genéticos tales como el *cruce* y la *mutación*. Como en todos los algoritmos evolutivos, es necesario diseñar una *función de adaptación* para cada problema que se desee resolver. Dado un cromosoma de la población, esta función devuelve un único valor numérico que se supone proporcional al grado de bondad de la solución que dicho cromosoma codifica. Esta función es la encargada de guiar al algoritmos genético por el espacio de búsqueda. Por esta razón, debe estar bien diseñada para que sea capaz, no sólo de distinguir de un modo claro los individuos bien adaptados de los que no lo están, sino también de ordenar éstos en función de su capacidad para resolver el problema.

El algoritmo que se presenta a continuación muestra la estructura general de un Algoritmos Genético básico, donde $P(t)$ denota la población en la generación t,

Algoritmo 1: Algoritmo Genético

```
Generar Población Inicial P0
Evaluar P0
t = 0
while no se cumpla el criterio de parada do
    Seleccionar Pt + 1 a partir de Pt
    Cruzar y mutar Pt+1
    Evaluar Pt+1
    t = t+1
end while
```

Una vez que cada individuo de la población inicial ha sido evaluado mediante la función de adaptación comienza el proceso iterativo representado en la Fig. A.1:

- **Selección**: es el mecanismo encargado de obtener una nueva población formada por copias de los mejores individuos de la población anterior, es decir, aquellos que obtienen un mejor valor en la función de adaptación.
- **Cruce**: este operador implica combinar dos individuos (padres) de la nueva población para generar dos nuevos individuos (descendientes) con la intención de que estos últimos, obtenidos mediante la recombinación de los primeros, estén mejor adaptados al problema. Este operador solo se aplica sobre una selección aleatoria de la nueva población, en función de un parámetro denominado probabilidad de cruce (Pc).
- **Mutación**: este segundo operador altera aleatoriamente, en función de una probabilidad de mutación (Pm), uno o más genes de los individuos de la nueva población. El objetivo del operador de mutación es explorar el espacio de búsqueda, ya que el cambio aleatorio suele conllevar un salto a otra zona del es

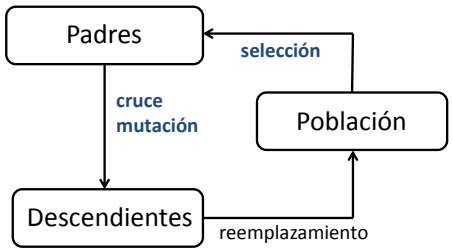

Figura A.1: Proceso Iterativo de un Algoritmo Genético

Al conjunto de individuos generados en cada uno de los ciclos de evolución del algoritmo se le conoce con el nombre de generación. El proceso iterativo finaliza cuando se cumple el criterio de parada.

De forma adicional un algoritmo genético puede ser completado con el concepto de elitismo, basado en mantener entre generaciones algunos de los individuos mejor evaluados.

Apéndice B

Algoritmos Evolutivos Multi-Objetivos

Se conoce con el nombre de Algoritmos Evolutivos Multi-Objetivo (AEMOs) a aquellos algoritmos genéticos en los que se definen múltiples objetivos relevantes para un mismo problema que, en general, están en conflicto. Es necesario por tanto definir múltiples funciones de adaptación a evaluar, cada una de ellas asociada a un objetivo diferente.

La mejor forma de resolver este tipo de problemas es mediante el uso de los criterios de dominancia y pareto-optimalidad. Una solución domina a otra si es mejor o igual en todos los objetivos y al menos mejor en uno de ellos. De esta forma, todas las soluciones que no son dominadas por ninguna otra solución se llaman pareto-optimales o no-dominadas. No suele existir una única solución optimal, existe un conjunto (a veces infinito) de soluciones no-dominadas que forma la frontera o frente de Pareto.

En los últimos años han habido avances significativos en el desarrollo de algoritmos evolutivos para problemas de optimización multi-objetivos. Los AEMOs tratan simultáneamente con un conjunto de posibles soluciones (llamado población), permitiéndoles encontrar varios miembros del conjunto optimal de Pareto en una sola ejecución del algoritmo. Además, estos algoritmos no son demasiado susceptibles por la forma o la continuidad del frente de Pareto (por ejemplo, pueden tratar fácilmente con frentes de Pareto discontinuos y cóncavos) [SR11].

El primer indicio respecto a la posibilidad de usar un algoritmos evolutivos para resolver un problema multi-objetivo aparece en tesis doctoral de 1967 [Ros67], aunque en una tesis no se desarrolló un AEMO actual porque el problema multi-objetivo fue formulado como un problema de un solo objetivo y resuelto con un algoritmo genético. David Schaffer es considerado el primer investigador en haber diseñado un AEMO a mediados de los años ochenta [Sch85]. El enfoque de Schaffer, llamado *Evaluated Genetic Algorithm* (VEGA) está basado en un algoritmo genético simple con un mecanismo de selección modificado. Sin embargo, VEGA presenta una serie de problemas donde el más importante es su incapacidad para retener soluciones con buena calidad, tal vez por encima de la media, pero no excepcional para alguno de los objetivos.

Después de VEGA, fue diseñada la primera generación de AEMOs caracterizada por su simplicidad, donde la principal lección aprendida fue que los AEMOs debían combinar un buen mecanismo para seleccionar los individuos no dominados (no necesariamente basado en el concepto de Pareto óptimo) y un buen mecanismo para mantener la diversidad. Los AEMOs más representativos de esta generación son: NSGA [SD94], NPGA [HNG94] y MOGA [FF93].

La segunda generación de los AEMOs comenzó cuando el elitismo llegó a ser en un mecanismo estándar. De hecho, el uso del elitismo es un requisito teórico para garantizar la convergencia de los AEMOs. Muchos AEMOs han sido propuestos en la segunda generación, la cual todavía estamos viviendo, pero muy pocos han sido considerados como un referente. SPEA2 [ZLT01] y el NSGA-II [DAPM02] pueden ser los AEMOs más representativos de esta generación. Además, en la actualidad ha crecido el interés por los AEMOs basados en descomposición (MOEA/D [ZL07] y MOEA/D-DE [LZ09]).

A continuación describimos el funcionamiento de dos de los AEMOs más populares: NSGA-II [DAPM02] y MOEA/D-DE [LZ09].

B.1. NSGA-II

Propuesto por Deb y otros en el año 2002 [DAPM02] es uno de los AEMOs más utilizados por la comunidad científica. Su funcionamiento se basa en el concepto de no dominancia para realizar una clasificación de la población por frentes. Así los individuos que pertenecen al primer frente son los no dominados; los que pertenecen al segundo frente son los no dominados en ausencia de los del frente

anterior, y así sucesivamente. A cada individuo dentro de cada frente se le asigna un rango equivalente a su nivel de no dominancia. Los mejores individuos son aquellos que tienen rangos menores y, por tanto, más posibilidades de reproducirse en la siguiente generación.

Este AEMO incorpora también el cálculo de una distancia de "crowding" que va a permitir mantener la diversidad de la población, con el fin de mejorar la selección por torneo binario. A continuación se describe su esquema de funcionamiento.

Algoritmo 2: Esquema del algoritmo NSGA-II

Entradas: N (tamaño de la población), T (máximo número de generaciones)
Salida: Q (conjunto de soluciones no dominadas)
Generar población inicial P_0 de tamaño N
A cada individuo se le asigna un objetivo igual a su nivel de no dominancia
Los operadores de selección, cruce y mutación se utilizan para crear la población de descendientes Q_0 de tamaño N
repeat
  $R_t = P_t \bigcup Q_t$
  $F =$ ordenar R_t según la no dominancia
  $P_{t+1} = 0$ y $i = 1$
  while $|P_{t+1}| + |F_i| \leqq N$ **do**
    Calcular la distancia de cruce en F_i
    $P_{t+1} = P_{t+1} \bigcup F_i$
    $i = i + 1$
  end while
  Ordenar F_i de forma descendiente utilizando el operador de "crowding"
  Seleccionar los primeros (N - $|P_{t+1}|$) elementos de F_i
  Usar los operadores de selección, cruce y mutación sobre P_{t+1} para generar la nueva población Q_{t+1}
  $t = t + 1$
until se alcanza el criterio de parada, $t > T$

B.2. MOEA/D-DE

Propuesto por Li y otros en el año 2009 [LZ09] representa un nueva versión del algoritmo MOEA/D (*Multiobjective Evolutionary Algorithm Based on Decomposition*) [ZL07] basada en la evolución diferencial (DE). Este algoritmo descompone el problema de optimización multi-objetivo en N subproblemas de optimización escalares con el propósito de cada uno de estos subproblemas optimice una agregación distinta de todos los objetivos. Las relaciones de vecindad entre estos subproblemas se definen sobre la base de las distancias entre los vector de pesos de cada subproblema. A continuación se describe su esquema de funcionamiento.

Algoritmo 3: Esquema del algoritmo MOEA/D-DE

Entradas: N (tamaño de la población), T (máximo número de generaciones), $\lambda^1, ...\lambda^N$ (conjunto de N vectores de pesos), T (número de vectores en el conjunto de vecinos de cada vector de pesos), δ (probabilidad de que las soluciones padres sean seleccionadas de la vecindad), η_r (número máximo de soluciones reemplazadas por cada solución hijo)

Salida: PS (soluciones no dominadas), PF (vectores objetivo de las soluciones no dominadas)

Inicialización:

Crear la vecindad de cada vector λ^i con los T vectores de pesos más cercanos.

Generar población inicial P_0 de tamaño N

Inicializar el vector de puntos de referencia $z = (z_1, ..., z_m)$ asignando $z_j = max_{1 \leq i \leq N} f_j(x^i)$, donde $j = 1, ..., m$

repeat

  for $i = 0$ to N **do**

    Generar un valor aleatorio $rand$ en el intervalo $[0, 1]$.

    if $rand < \delta$ **then**

      P = $B(i)$

    else

      P = población completa

    end if

    Asignar $r_1 = i$ y seleccionar aleatoriamente r_2 y r_3 de P y generar una solución hija y a partir de x^{r_1}, x^{r_2} y x^{r_3} aplicando el operador DE, el operador de mutación y el de reparación.

    Evaluar y

    for $j = 1$ to m **do**

      if $z_j < f_j(y_k)$ **then**

        $z_j = f_j(y_k)$

      end if

    end for

    while $c \leq \eta_r$ y P no está vacía **do**

      Seleccionar al azar un índice l de P.

      if $g(y_k|\lambda^l, z) \leq g(x^l|\lambda^l, z)$ **then**

        $x^l = y_k$

        $c = c + 1$

        Eliminar l de P

      end if

    end while

  end for

until se alcanza el criterio de parada, $t > T$

Apéndice C

Resultados Obtenidos en el Estudio Experimental

Este último apéndice contiene los resultados obtenidos en los estudios experimentales desarrollados a lo largo de esta memoria para evaluar la efectividad de los métodos presentados sobre 26 BDs. Este apéndice se divide en tres secciones para mostrar los resultados utilizados en la evaluación de los métodos QAR-CIP-NSGA-II, MOPNAR y NIGAR, respectivamente.

C.1. Resultados obtenidos para evaluar el método QAR-CIP-NSGA-II

Tabla C.1: Resultados obtenidos en la comparación con el clásico NSGA-II

Algoritmos	*#R*	Med_{Sop}	Med_{Conf}	Med_{Lift}	Med_{Conv}	Med_{FC}	$Med_{Netconf}$	$Med_{Yule'sQ}$	Med_{Amp}	$Med_{HiperVol}$	*%Ejem*
Balance											
QAR-CIP-NSGA-II_*C*	26,20	**0,18**	0,76	2,23	∞	0,57	0,44	0,72	**2,35**	3,80	82,08
QAR-CIP-NSGA-II	**132,20**	0,04	**0,92**	**6,37**	∞	**0,88**	**0,68**	**0,93**	3,97	**8,89**	**91,72**
Basketball											
QAR-CIP-NSGA-II_*C*	44	**0,23**	0,88	26,08	∞	0,73	0,63	0,85	**2**	97,59	84,79
QAR-CIP-NSGA-II	**200,60**	0,03	**0,99**	**82,09**	∞	**0,98**	**0,96**	**1**	2,10	**129,66**	**96,67**
Bolts											
QAR-CIP-NSGA-II_*C*	65	**0,19**	0,99	12,52	∞	0,98	0,94	**1**	2,26	51,17	81,50
QAR-CIP-NSGA-II	**174**	0,08	**1**	**30,19**	∞	**1**	**0,99**	**1**	**2,10**	**58,19**	**100**
Coil2000											
QAR-CIP-NSGA-II_*C*	**81,60**	**0,41**	0,88	27,43	∞	0,71	0,51	**0,86**	4,18	**3831,58**	**100**
QAR-CIP-NSGAII	58,60	0,26	**0,93**	**127,77**	∞	**0,93**	**0,80**	**0,86**	**3,33**	3622,53	**100**
House16H											
QAR-CIP-NSGA-II_*C*	100	**0,45**	0,92	369,31	∞	0,80	0,59	**0,87**	3,77	**42289,10**	**99,89**
QAR-CIP-NSGA-II	**288,20**	0,18	**0,93**	**2549,17**	∞	**0,91**	**0,71**	0,84	**2,98**	42008,83	99,81
Ionosphere											
QAR-CIP-NSGA-II_*C*	100	**0,25**	0,91	15,08	∞	0,85	0,76	0,97	3,08	**629,47**	84,45
QAR-CIP-NSGAII	**125,20**	0,10	**0,96**	**143,56**	∞	**0,94**	**0,90**	**0,99**	**2,67**	628,28	**87,70**
Letter											
QAR-CIP-NSGA-II_*C*	85,40	**0,28**	**0,88**	14,74	∞	0,77	0,63	**0,91**	**3,32**	474,669	91,95
QAR-CIP-NSGAII	**94,60**	0,08	0,87	**344,34**	∞	**0,84**	**0,68**	0,83	3,62	**5215,05**	**97,03**
Magic											
QAR-CIP-NSGA-II_*C*	100	**0,25**	0,91	15,08	∞	0,85	0,76	**0,93**	3,08	**34240,53**	84,45
QAR-CIP-NSGAII	**125,20**	0,10	**0,96**	**143,56**	∞	**0,94**	**0,90**	0,71	**2,67**	33861,74	**87,70**
Movement Libras											
QAR-CIP-NSGA-II_*C*	47	**0,09**	0,91	100,31	∞	0,90	0,89	**1**	**2,11**	408,40	45,51
QAR-CIP-NSGAII	**57,80**	0,05	**0,96**	**154,26**	∞	**0,96**	**0,94**	**1**	2,37	**502,22**	**55,28**
Optdigits											
QAR-CIP-NSGA-II_*C*	**100**	**0,28**	**0,86**	**50,83**	∞	0,74	0,54	**0,85**	3,70	**2697,22**	**100**
QAR-CIP-NSGAII	85,80	0,20	0,85	35,70	∞	**0,81**	**0,56**	0,84	**3,48**	1866,84	**100**
Penbased											
QAR-CIP-NSGA-II_*C*	**100,00**	**0,18**	**0,89**	43,20	∞	0,84	**0,67**	**0,89**	3,49	1324,24	66,26
QAR-CIP-NSGAII	91,20	0,07	0,88	**164,75**	∞	**0,86**	0,66	0,81	**3,23**	**2074,16**	**93,63**
Pollution											
QAR-CIP-NSGA-II_*C*	46,40	**0,23**	0,95	28,08	∞	0,91	0,87	0,99	**2**	61,66	83,67
QAR-CIP-NSGA-II	**249,40**	0,05	**1**	**51,75**	∞	**0,99**	**0,97**	**1**	2,09	**82,58**	**100**
Quake											
QAR-CIP-NSGA-II_*C*	100	**0,27**	0,90	72,81	∞	0,77	0,60	**0,85**	2,55	3002,71	**96,32**
QAR-CIP-NSGA-II	**137,20**	0,08	**0,93**	**450,67**	∞	**0,89**	**0,62**	0,77	**2,32**	**3318,70**	71,60
Satimage											
QAR-CIP-NSGA-II_*C*	100	**0,37**	0,92	24,07	∞	0,86	0,69	0,93	**3,81**	610,74	**100**
QAR-CIP-NSGAII	**299,80**	0,29	**0,93**	**34,04**	∞	**0,90**	**0,78**	**0,96**	5,26	**1746,54**	**100**
Segment											
QAR-CIP-NSGA-II_*C*	100	**0,38**	0,97	72,07	∞	0,87	**0,75**	**0,95**	2,72	3647,37	99,35
QAR-CIP-NSGAII	**176,60**	0,18	**1**	**451,22**	∞	**0,99**	0,71	0,79	**2,46**	**3923,47**	**100**
Sonar											
QAR-CIP-NSGA-II_*C*	69	**0,26**	0,83	14,25	∞	0,71	0,56	0,88	2,74	308,93	**85,68**
QAR-CIP-NSGAII	**123**	0,06	**0,95**	**128,39**	∞	**0,92**	**0,87**	**0,97**	**2,38**	**336,64**	80,20
Spambase											
QAR-CIP-NSGA-II_*C*	100	**0,47**	**0,92**	46,73	∞	0,75	0,52	0,9	4,92	3859,40	99,99
QAR-CIP-NSGAII	**178,60**	0,29	**0,92**	**154,34**	∞	**0,86**	**0,62**	**0,91**	**4,50**	**6339,77**	**100**
Spectfheart											
QAR-CIP-NSGA-II_*C*	**100**	**0,34**	**0,90**	6,93	∞	0,78	0,59	0,92	3,7	476,34	89
QAR-CIP-NSGAII	89,60	0,19	0,88	**34,61**	∞	**0,80**	**0,65**	**0,93**	**3,15**	**484,55**	**95,66**
Stock											
QAR-CIP-NSGA-II_*C*	88,20	**0,18**	0,93	35,57	∞	0,91	0,87	**1**	**2,83**	1267,15	55,04
QAR-CIP-NSGA-II	**107,60**	0,08	**0,94**	**91,91**	∞	**0,93**	**0,90**	**1**	2,97	**1456,81**	**73,75**
Stulong											
QAR-CIP-NSGA-II_*C*	100	**0,47**	**0,90**	**47,16**	∞	0,71	0,51	0,9	2,99	2345,82	**99,99**
QAR-CIP-NSGA-II	**153,80**	0,19	0,82	39,32	∞	**0,74**	**0,58**	**0,92**	**2,90**	**2414,68**	99,94
Texture											
QAR-CIP-NSGA-II_*C*	86	**0,27**	0,95	137,33	∞	0,93	**0,83**	**0,96**	2,93	8512,86	95,50
QAR-CIP-NSGAII	**151**	0,14	**0,98**	**785,39**	∞	**0,97**	0,79	0,84	**2,66**	**9294,31**	**99,87**
Thyroid											
QAR-CIP-NSGA-II_*C*	100,00	**0,51**	**0,93**	58,24	∞	0,76	0,48	0,84	3,60	4912,50	99,99
QAR-CIP-NSGAII	**216,60**	0,29	0,93	**163,67**	∞	**0,88**	**0,63**	**0,92**	**3,30**	**9168,67**	**100**
Vehicle											
QAR-CIP-NSGA-II_*C*	74,40	**0,29**	0,97	40,14	∞	0,91	0,86	0,98	**2,39**	1092,15	90,36
QAR-CIP-NSGAII	**98,40**	0,15	**0,98**	**88,50**	∞	**0,98**	**0,95**	**1,00**	2,41	**1262,27**	**99,91**
Vowel											
QAR-CIP-NSGA-II_*C*	55,80	**0,17**	0,81	162,79	∞	0,68	0,63	0,87	**2,45**	1310,20	**100**
QAR-CIP-NSGAII	**84,40**	0,03	**0,95**	**540,52**	∞	**0,95**	**0,92**	**0,99**	2,48	**1524,08**	84,19
Wdbc											
QAR-CIP-NSGA-II_*C*	87,20	**0,35**	0,92	23,76	∞	0,84	0,75	0,97	2,93	943,73	91,15
QAR-CIP-NSGAII	**126,80**	0,17	**0,99**	**220,99**	∞	**0,98**	**0,96**	**1**	**2,26**	**957,10**	**98,57**
Wine											
QAR-CIP-NSGA-II_*C*	61,40	**0,17**	0,91	42,60	∞	0,85	0,78	0,96	2,37	235,12	**84,61**
QAR-CIP-NSGAII	**122,20**	0,04	**0,97**	**118,50**	∞	**0,97**	**0,95**	**1**	**2,31**	**269,07**	83,94

Tabla C.2: Resultados para las BDs Balance, Basketball, Bolts, Coil2000 y House16H en la comparación con los algoritmos evolutivos

Algoritmos	*#R*	Med_{Sop}	Med_{Conf}	Med_{Lift}	Med_{Conv}	Med_{FC}	$Med_{Netconf}$	$Med_{Yule'sQ}$	Med_{Amp}	*%Ejem*
Balance										
EARMGA	100	0,49	**1**	1	1	0	0	0	**2**	**100**
GAR	0	-	-	-	-	-	-	-	-	-
GENAR	30	0,13	0,94	2,05	30,95	**0,89**	0,56	0,92	5	85,13
Alatasetal	24,2	0,31	**1**	1,1	∞	0,05	0,03	0,05	4,36	**100**
ARMMGA	1	0,28	0,68	1,47	1,67	0,4	0,36	0,64	**2**	27,04
MODENAR	38,4	0,13	0,64	1,38	∞	0,32	0,18	0,39	3,04	75,43
MOEA_Ghosh	24,6	**0,91**	0,95	1	1	0	0	0	4,32	**100**
QAR-CIP-NSGA-II	**132,2**	0,04	0,92	**6,37**	∞	0,88	**0,68**	**0,93**	3,97	91,72
Basketball										
EARMGA	100	0,24	**1**	1,05	∞	0,18	0,03	0,12	**2**	**100**
GAR	2	0,77	0,89	1,02	1,13	0,11	0,11	0,37	**2**	97,09
GENAR	30	0,3	0,97	1,1	∞	0,68	0,12	0,64	5	89,8
Alatasetal	8,6	**0,98**	**1**	1	1	-0,01	-0,01	0	3,2	**100**
ARMMGA	1	0,23	0,85	1,73	3,32	0,7	0,49	0,82	**2**	22,92
MODENAR	61,4	0,32	0,79	2,37	∞	0,33	0,15	2,3	97,5	
MOEA_Ghosh	25,8	0,9	0,98	2,55	∞	0,14	0,07	0,24	3,57	**100**
QAR-CIP-NSGA-II	**200,6**	0,03	0,99	**82,09**	∞	**0,98**	**0,96**	**1**	2,1	96,67
Bolts										
EARMGA	100	0,35	**1**	1,1	∞	0,2	0,07	0,2	**2**	**100**
GAR	33,2	0,21	0,98	4,21	∞	0,96	0,86	0,99	3,36	91,5
GENAR	30	0,14	**1**	1,57	∞	**1**	0,42	**1**	8	39
Alatasetal	21	**0,95**	**1**	1,04	∞	0,14	0,14	0,14	3,65	95
ARMMGA	1	0,46	**1**	1,3	∞	**1**	0,42	**1**	**2**	46
MODENAR	39,6	0,39	0,94	2,27	∞	0,48	0,41	0,54	3,93	68,5
MOEA_Ghosh	11,8	0,76	0,95	4,08	∞	0,33	0,29	0,37	6,33	**100**
QAR-CIP-NSGA-II	**174**	0,08	**1**	**30,19**	∞	**1**	**0,99**	**1**	2,1	**100**
Coil2000										
EARMGA	72,40	0,33	**1**	1,01	∞	0,11	0,01	0,01	**2**	**100**
GAR	**197,60**	0,94	0,98	1,01	∞	0,06	0,03	0,08	2,07	**100**
GENAR	30	0,01	0,96	1,02	∞	0,37	0,02	0	86	13,74
Alatasetal	0	-	-	-	-	-	-	-	-	-
ARMMGA	1	**1**	**1**	1	∞	0,39	0,07	0	**2**	99,78
MODENAR	35,60	0,05	0,69	510,26	∞	0,52	0,26	0,47	53,07	25,88
MOEA_Ghosh	49,60	0,01	0,33	**725,26**	∞	0,33	0,18	0,32	72,08	0,28
QAR-CIP-NSGAII	58,60	0,26	0,93	127,77	∞	**0,93**	**0,80**	**0,86**	3,33	**100**
House16H										
EARMGA	60,2	0,17	**1**	1,01	∞	0,28	0,01	0,01	3	98,56
GAR	105,6	0,76	0,9	1,03	1,33	0,2	0,17	0,48	**2,01**	**99,99**
GENAR	30	0,44	0,99	1,02	2,09	0,44	0,03	0,41	17	87,31
Alatasetal	89,67	0,19	0,99	1,03	∞	0,58	0,03	0,41	8,77	98,09
ARMMGA	1	**0,98**	**1**	1,01	1,13	0,09	0,04	0,16	3	97,97
MODENAR	64,4	0,69	0,99	1,15	∞	0,72	0,19	0,74	7	81
MOEA_Ghosh	19,8	0,6	0,79	269,17	∞	0,36	0,11	0,3	7,75	98,87
QAR-CIP-NSGA-II	**288,2**	0,18	0,93	**2549,17**	∞	**0,91**	**0,71**	**0,84**	2,98	99,81

Tabla C.3: Resultados para las BDs Ionosphere, Letter, Magic, Movement Libras, Optdigits y Pollution en la comparación entre los algoritmos evolutivos y QAR-CIP-NSGA-II

Algoritmos	*#R*	Med_{Sop}	Med_{Conf}	Med_{Lift}	Med_{Conv}	Med_{FC}	$Med_{Netconf}$	$Med_{Yule'sQ}$	Med_{Amp}	*%Ejem*
Ionosphere										
EARMGA	100	0,40	**1**	1,01	∞	0,01	0	0,01	**2**	**100**
GAR	39	0,21	0,93	1,62	∞	0,80	0,42	0,83	**2**	77,33
GENAR	30	0,30	0,99	1,55	∞	0,97	0,50	0,98	34	35,22
Alatasetal	84	0,02	**1**	11,16	∞	**1**	0,85	**1**	9,39	3,14
ARMMGA	1	**0,54**	0,87	1,41	3,02	0,65	0,64	0,92	**2**	53
MODENAR	28	0,26	0,76	42,04	∞	0,59	0,65	0,83	22,09	55,28
MOEA_Ghosh	**292**	0,01	0,07	21,08	∞	0,06	0,06	0,13	31,59	1,26
QAR-CIP-NSGAII	125,20	0,10	0,96	**143,56**	∞	0,94	**0,90**	0,99	2,67	87,70
Letter										
EARMGA	**100**	0,35	**1**	1	∞	0,01	0	0	**2**	**100**
GAR	12,2	**0,58**	0,85	1,19	1,7	0,34	0,29	0,56	2,02	99,79
GENAR	30	0,02	0,32	8,15	2,36	0,29	0,3	**0,84**	17	67,99
Alatasetal	25,33	0,2	0,99	4,02	∞	0,75	0,34	0,23	7,17	51,65
ARMMGA	1	0,49	0,82	1,13	1,65	0,3	0,21	0,42	**2**	49,28
MODENAR	56,6	0,33	0,95	1,44	∞	0,67	0,06	0,58	7,2	90,46
MOEA_Ghosh	29	0,48	0,82	58,52	∞	0,35	0,2	0,37	9,6	97,33
QAR-CIP-NSGAII	94,6	0,08	0,87	**344,34**	∞	**0,84**	**0,68**	0,83	3,62	97,03
Magic										
EARMGA	92	0,31	**1**	1,05	∞	0,05	0,02	0,03	**2**	**100**
GAR	57,80	0,67	0,91	1,10	2,33	0,48	0,35	0,74	2,08	96,84
GENAR	30	0,43	0,82	1,26	1,9	0,47	0,35	0,67	11	62,56
Alatasetal	11	0,47	**1**	1,26	956,59	0,88	0,34	**0,91**	4,73	89,9
ARMMGA	1	**0,87**	0,96	1,05	9,08	0,47	0,41	0,89	**2**	86,31
MODENAR	73,60	0,46	0,94	234,06	∞	0,50	0,05	0,48	3,80	80,31
MOEA_Ghosh	29,40	0,72	0,90	1,46	∞	0,52	0,17	0,5	5,26	98,33
QAR-CIP-NSGAII	**210,80**	0,22	0,95	**4464,87**	∞	**0,93**	**0,56**	0,71	2,35	99,95
Movement Libras										
EARMGA	**100**	0,39	**1**	1	∞	0	0	0	**2**	**100**
GAR	2,60	**0,42**	0,94	3,73	12,62	0,90	0,90	**1**	**2**	53,28
GENAR	30	0,04	0,89	13,35	∞	0,89	0,86	0,99	91	53,73
Alatasetal	0	-	-	-	-	-	-	-	-	-
ARMMGA	1	0,26	0,87	3,27	28,86	0,79	0,79	0,93	**2**	25,95
MODENAR	23,40	0,01	0,15	32,31	∞	0,04	0,15	0,17	61,61	3,17
MOEA_Ghosh	10,80	0,01	0,22	79,47	∞	0,22	0,22	0,22	80,08	0,28
QAR-CIP-NSGAII	57,80	0,05	0,96	**154,26**	∞	**0,96**	**0,94**	**1**	2,37	55,28
Optdigits										
EARMGA	**89**	0,41	**1**	1,01	∞	0,04	0,01	0	**2**	**100**
GAR	65,8	0,71	0,97	1,02	∞	0,18	0,03	0,04	2,01	**100**
GENAR	16,2	0,01	**1**	10,1	∞	**1**	**0,91**	**0,86**	65	1,78
Alatasetal	0	-	-	-	-	-	-	-	-	-
ARMMGA	1	**1**	**1**	1	1	0	0	0	**2**	99,98
MODENAR	8	0,01	0,69	**3668,12**	∞	0,52	0,01	0,02	44,48	0,20
MOEA_Ghosh	47,80	0,01	0,16	859,53	∞	0,16	0,01	0	54,8	0,05
QAR-CIP-NSGAII	85,80	0,20	0,85	35,70	∞	0,81	0,56	0,84	3,48	**100**
Penbased										
EARMGA	**100**	0,36	**1**	1	1	0	0	0	**2**	**100**
GAR	2	**0,72**	0,87	1,04	1,22	0,18	0,18	0,48	2	94,95
GENAR	30	0,05	0,97	9,6	∞	**0,97**	**0,91**	**1**	17	44,79
Alatasetal	0	-	-	-	-	-	-	-	-	-
ARMMGA	1	0,42	0,8	1,36	3	0,5	0,41	0,71	2	41,4
MODENAR	68,2	0,25	0,95	1,5	∞	0,84	0,33	0,86	7,2	50,04
MOEA_Ghosh	24,4	0,57	0,75	62,26	∞	0,27	0,14	0,44	9,18	96,5
QAR-CIP-NSGAII	91,2	0,07	0,88	**164,75**	∞	0,86	0,66	0,81	3,23	93,63
Pollution										
EARMGA	100	0,25	**1**	1,19	∞	0,31	0,06	0,27	**2**	**100**
GAR	54	**0,67**	0,91	1,17	∞	0,53	0,43	0,77	2,03	**100**
GENAR	30	0,22	**1**	1,23	∞	0,98	0,24	0,98	16	48
Alatasetal	17,2	0,59	**1**	6,86	∞	0,45	0,39	0,45	3,29	59,67
ARMMGA	1	0,64	**1**	1,04	∞	**1**	0,1	**1**	**2**	63,34
MODENAR	34,4	0,27	0,91	2,94	∞	0,85	0,52	0,94	7,2	48,34
MOEA_Ghosh	26,8	0,18	0,67	8,89	∞	0,61	0,62	0,95	13,18	39
QAR-CIP-NSGA-II	**249,4**	0,05	**1**	**51,75**	∞	0,99	**0,97**	**1**	2,09	**100**

Tabla C.4: Resultados para las BDs Quake, Satimage, Segment, Sonar, Spambase, Spectfheart y Stock en la comparación con los algoritmos evolutivos

Algoritmos	*#R*	Med_{Sop}	Med_{Conf}	Med_{Lift}	Med_{Conv}	Med_{FC}	$Med_{Netconf}$	$Med_{Yule'sQ}$	Med_{Amp}	*%Ejem*
					Quake					
EARMGA	100	0,27	**1**	1	∞	0,01	0	0	**2**	**100**
GAR	1	0,44	0,84	0,98	0,88	-0,03	-0,05	-0,17	**2**	52,89
GENAR	30	0,55	0,95	1,01	1,09	0,09	0,02	0,1	4	81,78
Alatasetal	4,25	0,67	**1**	1,01	∞	0,1	0	0	2,08	98,06
ARMMGA	1	0,66	0,73	1,01	1,02	0,02	0,04	0,09	**2**	65,94
MODENAR	63,6	0,36	0,84	117,09	∞	0,31	0,09	0,22	2,09	92,84
MOEA_Ghosh	8	**0,86**	**1**	1,01	∞	0,18	0,01	0,14	3,09	100
QAR-CIP-NSGA-II	**137,2**	0,08	0,93	**450,67**	∞	**0,89**	**0,62**	**0,77**	2,32	71,6
					Satimage					
EARMGA	88	0,34	**1**	1,06	∞	0,03	0,02	0,02	**2**	**100**
GAR	207,80	0,91	0,97	1,04	2,28	0,40	0,38	0,77	2,13	**100**
GENAR	30	0,22	0,32	1,46	1,18	0,11	0,30	0,96	37	99,98
Alatasetal	0	-	-	-	-	-	-	-	-	-
ARMMGA	1	**0,96**	0,99	1,02	2,25	0,56	0,54	**0,98**	**2**	95,44
MODENAR	48,60	0,53	0,98	1,25	∞	**0,93**	0,34	0,94	18,60	88,65
MOEA_Ghosh	**708**	0,01	0,01	23,37	∞	0,01	0,01	0	36,90	0,06
QAR-CIP-NSGAII	299,80	0,29	0,93	**34,04**	∞	0,90	**0,78**	0,96	5,26	**100**
					Segment					
EARMGA	99,20	0,45	**1**	1,04	∞	0,08	0,02	0,04	**2**	**100**
GAR	18,80	0,36	0,89	2,49	3,61	0,58	0,47	0,73	**2**	97,97
GENAR	30	0,07	0,78	5,43	∞	0,74	0,70	0,93	20	83,49
Alatasetal	47	0,53	0,94	1,03	∞	0,26	0,02	0,21	4,15	**100**
ARMMGA	1	**0,92**	**1**	1,14	∞	0,20	0,20	0,20	**2**	91,33
MODENAR	58,80	0,33	0,97	1,72	∞	0,93	0,58	**0,96**	10,60	56,49
MOEA_Ghosh	28,20	0,36	0,86	108,6	∞	0,73	0,50	0,8	12,63	72,95
QAR-CIP-NSGAII	**176,60**	0,18	**1**	**451,22**	∞	**0,99**	**0,71**	0,76	2,46	**100**
					Sonar					
EARMGA	100	0,27	**1**	1,52	∞	0,13	0,02	0,08	**2**	**100**
GAR	9,60	0,41	0,84	1,33	2,16	0,46	0,31	0,62	**2**	93,27
GENAR	30	0,04	0,92	1,77	∞	0,83	0,42	0,83	61	30,10
Alatasetal	0	-	-	-	-	-	-	-	-	-
ARMMGA	1	**0,72**	0,91	1,05	1,61	0,28	0,22	0,56	**2**	72,02
MODENAR	14,80	0,01	0,25	52,63	∞	0,20	0,25	0,25	41,24	1,74
MOEA_Ghosh	26,40	0,01	0,35	72,01	∞	0,35	0,35	0,35	54,69	0,49
QAR-CIP-NSGAII	**123**	0,06	0,95	**128,39**	∞	**0,92**	**0,87**	**0,97**	2,38	80,20
					Spambase					
EARMGA	54,6	0,27	**1**	1,01	∞	0,39	0,01	0	**2**	80,85
GAR	1,50	0,22	0,94	1,75	7,69	0,84	0,44	0,88	**2**	27,40
GENAR	30	0,44	0,62	1,04	1,05	0,05	0,06	0,12	58	84,80
Alatasetal	0	-	-	-	-	-	-	-	-	-
ARMMGA	1	**1**	**1**	1	1	-0,01	-0,01	0	**2**	99,81
MODENAR	54	0,66	0,94	128,40	∞	0,39	0,10	0,5	36,4	98,83
MOEA_Ghosh	81	0,17	0,88	81,67	∞	0,81	0,61	0,85	37,47	31,51
QAR-CIP-NSGAII	**178,60**	0,29	0,92	**154,34**	∞	**0,86**	**0,62**	**0,91**	4,50	**100**
					Spectfheart					
EARMGA	100	0,36	**1**	1,01	∞	0,02	0,01	0,02	**2**	**100**
GAR	30	0,7	0,88	1,08	1,63	0,33	0,3	0,65	**2**	99,85
GENAR	30	0,25	0,68	0,91	0,68	-0,13	-0,17	-0,45	45	60,15
Alatasetal	0	-	-	-	-	-	-	-	-	-
ARMMGA	1	**0,74**	0,94	1,12	2,64	0,55	0,4	0,84	**2**	73,64
MODENAR	47	0,21	0,89	1,78	∞	0,77	0,48	0,86	22,6	50,79
MOEA_Ghosh	**458,6**	0,01	0,05	11,83	∞	0,05	0,05	0,07	43,65	0,6
QAR-CIP-NSGAII	89,6	0,19	0,88	**34,61**	∞	**0,8**	**0,65**	**0,93**	3,15	95,66
					Stock					
EARMGA	100	0,37	**1**	1,01	∞	0,02	0,01	0,02	**2**	**100**
GAR	2	0,56	0,87	1,35	3,19	0,62	0,62	0,88	**2**	73,3
GENAR	30	0,29	0,92	1,69	∞	0,81	0,54	0,89	10	88,51
Alatasetal	10,8	0,08	**1**	**96,75**	∞	0,91	0,72	0,73	2,73	21,04
ARMMGA	1	0,37	0,77	1,63	2,25	0,56	0,56	0,86	**2**	36,22
MODENAR	60,75	0,46	0,94	1,9	∞	0,75	0,38	0,54	3,25	77,58
MOEA_Ghosh	19,8	**0,61**	0,91	42,56	∞	0,53	0,36	0,68	5,28	96,4

Tabla C.5: Resultados para las BDs Stulong, Texture, Thyroid, Vehicle, Vowel, Wdbc y Wine en la comparación con los algoritmos evolutivos

Algoritmos	*#R*	Med_{Sop}	Med_{Conf}	Med_{Lift}	Med_{Conv}	Med_{FC}	$Med_{Netconf}$	$Med_{Yule'sQ}$	Med_{Amp}	*%Ejem*
QAR-CIP-NSGA-II	**107,6**	0,08	0,94	91,91	∞	**0,93**	**0,9**	**1**	2,97	73,75
					Stulong					
EARMGA	92,6	0,27	**1**	1,01	∞	0,13	0,01	0,02	**2**	**100**
GAR	157,4	0,78	0,94	1,03	1,63	0,31	0,21	0,63	2,96	99,94
GENAR	30	**0,88**	0,99	1,01	1,02	0,02	0,01	0,04	5	95,26
Alatasetal	7,33	0,72	0,99	1,5	∞	0,24	0,08	0,11	2,95	99,25
ARMMGA	1	0,87	0,87	1,01	1,03	0,03	**0,62**	0,91	**2**	86,38
MODENAR	63,2	0,52	0,88	13,94	∞	0,27	0,06	0,27	3	99,28
MOEA_Ghosh	19,6	0,83	0,99	1,04	∞	0,51	0,23	0,6	3,47	99,92
QAR-CIP-NSGA-II	**153,8**	0,19	0,82	**39,32**	∞	**0,74**	0,58	**0,92**	2,9	99,94
					Texture					
EARMGA	100	0,27	**1**	3,52	∞	0,08	0,05	0,07	**2**	**100**
GAR	38,80	**0,72**	0,94	1,23	∞	0,72	0,67	0,92	2,03	97,98
GENAR	30	0,09	0,69	7,51	∞	0,66	0,68	**0,99**	41	98,18
Alatasetal	0	-	-	-	-	-	-	-	-	-
ARMMGA	1	0,71	0,95	1,34	∞	0,86	**0,87**	0,98	**2**	70,42
MODENAR	29	0,07	0,48	4,42	∞	0,21	0,39	0,66	27,55	39,74
MOEA_Ghosh	81,40	0,01	0,17	**916,93**	∞	0,17	0,01	0	37,01	0,04
QAR-CIP-NSGAII	**151**	0,14	0,98	785,39	∞	**0,97**	0,79	0,84	2,66	99,87
					Thyroid					
EARMGA	80,2	0,56	**1**	1,01	∞	0,03	0,01	0	**2**	**100**
GAR	191,2	0,83	0,92	1,02	1,17	0,14	0,13	0,46	2,01	99,97
GENAR	30	0,59	0,93	1	1,01	0,03	0	-0,01	22	94,3
Alatasetal	80	0,32	0,96	1,02	∞	0,5	0,06	0,24	17,05	89,27
ARMMGA	1	**1**	1	**1**	1	-0,01	-0,01	0	**2**	99,08
MODENAR	71,4	0,72	0,99	1,01	∞	0,23	0,02	0,16	12,8	95,89
MOEA_Ghosh	28,8	0,99	0,99	1	1	0	0	0,02	15,45	**100**
QAR-CIP-NSGAII	**216,6**	0,29	0,93	**163,67**	∞	**0,88**	**0,63**	**0,92**	3,3	**100**
					Vehicle					
EARMGA	100	0,32	**1**	1,08	∞	0,11	0,04	0,09	**2**	**100**
GAR	24,8	0,67	0,94	1,31	∞	0,48	0,36	0,79	2,03	**100**
GENAR	30	0,09	0,67	2,62	2,6	0,56	0,48	0,76	19	66,39
Alatasetal	22,4	0,01	**1**	77,37	∞	**1**	0,79	0,61	4,27	0,24
ARMMGA	1,2	**0,74**	0,96	1,16	9,25	0,65	0,22	0,76	**2**	78,35
MODENAR	60,2	0,38	0,95	1,71	∞	0,88	0,61	0,95	8,6	65,16
MOEA_Ghosh	**470**	0,17	0,53	50,16	∞	0,49	0,39	0,73	15,92	31,07
QAR-CIP-NSGAII	98,4	0,15	0,98	**88,5**	∞	0,98	**0,95**	**1**	2,41	99,91
					Vowel					
EARMGA	**100**	0,34	**1**	1,01	∞	0,01	0,01	0,01	**2**	**100**
GAR	2	**0,77**	0,88	1,02	1,12	0,10	0,10	0,35	**2**	97,04
GENAR	30	0,02	0,54	5,93	∞	0,50	0,47	0,85	14	63,26
Alatasetal	89,60	0,13	**1**	63,92	∞	0,76	0,17	0,64	8,33	92,63
ARMMGA	1	0,26	0,95	1,99	∞	0,91	0,64	0,94	**2**	25,34
MODENAR	63,40	0,17	0,66	3,08	∞	0,46	0,40	0,58	5	29,60
MOEA_Ghosh	21	0,62	0,81	56,42	∞	0,24	0,12	0,33	6,53	99,94
QAR-CIP-NSGAII	84,40	0,03	0,95	**540,52**	∞	**0,95**	**0,92**	**0,99**	2,48	84,19
					Wdbc					
EARMGA	100	0,25	**1**	1,52	∞	0,17	0,06	0,1	**2**	**100**
GAR	88,80	0,54	0,86	1,19	2,49	0,37	0,32	0,49	**2**	98,84
GENAR	30	0,44	0,94	1,53	6,47	0,84	0,60	0,94	31	71,36
Alatasetal	0	-	-	-	-	-	-	-	-	-
ARMMGA	1	**0,94**	0,98	1,02	∞	0,46	0,21	0,73	**2**	93,47
MODENAR	37,80	0,19	0,77	35,57	∞	0,60	0,36	0,71	17,73	67,91
MOEA_Ghosh	**1338,20**	0,01	0,01	2,01	∞	0,01	0,01	0,01	30,90	0,57
QAR-CIP-NSGAII	126,80	0,17	0,99	**220,99**	∞	**0,98**	**0,96**	**1**	2,26	98,57
					Wine					
EARMGA	100	0,38	**1**	1,02	∞	0,05	0,01	0,05	**2**	**100**
GAR	8,20	0,22	0,98	2,79	∞	0,95	0,76	0,97	**2**	55,96
GENAR	30	0,19	**1**	3,02	∞	**1**	0,82	**1**	14	68,66
Alatasetal	28,20	0,27	**1**	25,89	∞	0,79	0,45	0,76	4,30	40,45
ARMMGA	1	0,26	0,85	2,70	8,53	0,75	0,72	0,92	**2**	25,06
MODENAR	54,40	0,39	0,91	6,07	∞	0,63	0,33	0,72	5,4	71,58
MOEA_Ghosh	22,40	**0,44**	0,85	10,37	∞	0,55	0,42	0,68	8,54	80,57
QAR-CIP-NSGAII	**122,20**	0,04	0,97	**118,50**	∞	0,97	**0,95**	**1**	2,31	83,94

Tabla C.6: Resultados para todas las BDs en la comparación entre los algoritmos clásicos de extracción de reglas de asociación y QAR-CIP-NSGA-II

Algoritmos	*#R*	Med_{Sop}	Med_{Conf}	Med_{Lift}	Med_{Conv}	Med_{FC}	$Med_{Netconf}$	$Med_{Yule'sQ}$	Med_{Amp}	*%Ejem*
Balance										
Apriori	2	**0,14**	0,86	6,25	5,77	0,83	**0,86**	**1**	**3**	27,20
Eclat	2	**0,14**	0,86	6,25	5,77	0,83	**0,86**	**1**	**3**	27,20
QAR-CIP-NSGA-II	**132,20**	0,04	**0,92**	**6,37**	∞	**0,88**	0,68	0,96	3,97	**91,72**
Basketball										
Apriori	4	**0,15**	0,87	4,88	6,65	0,84	0,81	0,99	2,75	33,34
Eclat	4	**0,15**	0,87	4,88	6,65	0,84	0,81	0,99	2,75	33,34
QAR-CIP-NSGA-II	**200,60**	0,03	**0,99**	**82,09**	∞	**0,98**	**0,96**	**1**	**2,10**	**97,67**
Bolts										
Apriori	**1246**	**0,15**	0,99	7,16	∞	0,98	0,96	**1**	4,36	97,50
Eclat	**1246**	**0,15**	0,99	7,16	∞	0,98	0,96	**1**	4,36	97,50
QAR-CIP-NSGA-II	174	0,08	**1**	**30,19**	∞	**1**	**0,99**	**1**	**2,10**	**100**
House16H										
Apriori	**1749917**	**0,22**	**0,97**	2,19	∞	0,83	0,45	0,88	8,65	**100**
Eclat	**1749917**	**0,22**	**0,97**	2,19	∞	0,83	0,45	0,88	8,65	**100**
QAR-CIP-NSGA-II	288,20	0,18	0,93	**2549,17**	∞	**0,91**	**0,71**	**0,92**	**2,98**	99,81
Ionosphere										
Apriori	**2,1e+08**	**0,12**	**0,98**	4,40	∞	**0,97**	0,72	**1**	10,79	**100**
Eclat	**2,1e+08**	**0,12**	**0,98**	4,40	∞	**0,97**	0,72	**1**	10,79	**100**
QAR-CIP-NSGAII	125,20	0,10	0,96	**143,56**	∞	0,94	**0,90**	0,99	**2,67**	87,70
Letter										
Apriori	**3916**	**0,14**	**0,88**	4,66	∞	0,81	**0,73**	**0,93**	4,55	**99,49**
Eclat	**3916**	**0,14**	**0,88**	4,66	∞	0,81	**0,73**	**0,93**	4,55	99,49
QAR-CIP-NSGAII	94,6	0,08	0,87	**344,34**	∞	**0,84**	0,68	0,83	**3,62**	97,03
Magic										
Apriori	**9785**	0,19	**0,96**	2,73	∞	0,87	0,52	**0,95**	5,53	**99,96**
Eclat	**9785**	0,19	**0,96**	2,73	∞	0,87	0,52	**0,95**	5,53	**99,96**
QAR-CIP-NSGAII	210,80	**0,22**	0,95	**4464,87**	∞	**0,93**	**0,56**	0,85	**2,35**	99,95
Penbased										
Apriori	**1918**	**0,14**	**0,92**	4,23	∞	0,82	0,63	**0,87**	4,07	**99,5**
Eclat	1918	0,14	0,92	4,23	∞	0,82	0,63	0,87	4,07	99,5
QAR-CIP-NSGAII	91,2	0,07	0,88	**164,75**	∞	**0,86**	**0,66**	0,81	**3,23**	93,63
Pollution										
Apriori	**41510**	**0,13**	0,95	5,84	∞	0,93	0,86	0,99	5,88	**100**
Eclat	**41510**	**0,13**	0,95	5,84	∞	0,93	0,86	0,99	5,88	**100**
QAR-CIP-NSGA-II	249,40	0,05	**1**	**51,75**	∞	**0,99**	**0,97**	**1**	**2,09**	**100**
Quake										
Apriori	18	**0,25**	0,91	1	1,15	0,11	-0,01	0,51	2,56	**90,55**
Eclat	18	**0,25**	0,91	1	1,15	0,11	-0,01	0,51	2,56	**90,55**
QAR-CIP-NSGA-II	**137,20**	0,08	**0,93**	**450,67**	∞	**0,89**	**0,62**	**0,88**	**2,32**	71,60
Stock										
Apriori	**855**	**0,13**	0,91	4,77	∞	0,88	0,76	0,98	4,16	**99,48**
Eclat	**855**	**0,13**	0,91	4,77	∞	0,88	0,76	0,98	4,16	**99,48**
QAR-CIP-NSGA-II	107,60	0,08	**0,94**	**91,91**	∞	**0,93**	**0,90**	**1**	**2,97**	73,75
Stulong										
Apriori	89	**0,31**	**0,93**	1,22	∞	0,43	0,14	0,64	3,26	99,86
Eclat	89	**0,31**	**0,93**	1,22	∞	0,43	0,14	0,64	3,26	99,86
QAR-CIP-NSGA-II	**153,60**	0,19	0,82	**39,32**	∞	**0,74**	**0,58**	**0,96**	**2,90**	**99,94**
Vehicle										
Apriori	**141107**	0,14	0,95	4,18	∞	0,91	0,76	0,95	6,4	**100**
Eclat	**141107**	0,14	0,95	4,18	∞	0,91	0,76	0,95	6,4	**100**
QAR-CIP-NSGAII	98,4	**0,15**	**0,98**	**88,5**	∞	**0,98**	**0,95**	**1**	**2,41**	99,91
Wine										
Apriori	**1348**	**0,13**	0,91	5,97	∞	0,87	0,82	0,98	4,07	**100**
Eclat	**1348**	**0,13**	0,91	5,97	∞	0,87	0,82	0,98	4,07	**100**
QAR-CIP-NSGAII	122,20	0,04	**0,97**	**118,50**	∞	**0,97**	**0,95**	**1**	**2,31**	83,94
Vowel										
Apriori	**235**	**0,14**	**0,98**	2,97	∞	**0,96**	0,69	0,98	3,48	**100**
Eclat	**235**	**0,14**	**0,98**	2,97	∞	**0,96**	0,69	0,98	3,48	**100**
QAR-CIP-NSGAII	84,40	0,03	0,95	**540,52**	∞	0,95	**0,92**	**0,99**	**2,48**	84,19

C.2. Resultados obtenidos para evaluar el método MOPNAR

Tabla C.7: Resultados obtenidos en la comparación entre Alatasetal y MOPNAR

Algoritmos	*#R*	Med_{Sop}	Med_{Conf}	Med_{Lift}	Med_{Conv}	Med_{FC}	$Med_{Netconf}$	$Med_{Yule'sQ}$	Med_{Amp}	*%Ejem*
Balance										
Alatasetal	24,2	**0,31**	**1**	1,1	∞	0,05	0,03	0,05	4,36	**100**
MOPNAR	**71**	0,15	0,83	**2,01**	∞	**0,71**	**0,43**	**0,81**	**2,95**	**100**
Basketball										
Alatasetal	8,6	**0,98**	**1**	1	1	-0,01	-0,01	0	3,2	**100**
MOPNAR	**91,6**	0,16	0,95	**47,73**	∞	**0,92**	**0,78**	**0,98**	**2,33**	99,79
Bolts										
Alatasetal	21	**0,95**	**1**	1,04	∞	0,14	0,14	0,14	3,65	95
MOPNAR	**52**	0,36	**1**	**14,16**	∞	**0,99**	**0,93**	**1**	**2,34**	**100**
Coil2000										
Alatasetal	0	-	-	-	-	-	-	-	-	-
MOPNAR	**28,4**	**0,36**	**0,88**	**5,41**	∞	**0,84**	**0,67**	**0,96**	**3,17**	**99,72**
House16H										
Alatasetal	89,67	0,19	**0,99**	1,03	∞	0,58	0,03	0,41	8,77	98,09
MOPNAR	**91**	**0,32**	0,88	**6,08**	∞	**0,84**	**0,65**	**0,97**	**2,99**	**99,94**
Ionosphere										
Alatasetal	**84**	0,02	**1**	11,16	∞	**1**	**0,85**	**1**	9,39	3,14
MOPNAR	72,8	**0,3**	0,93	**13,91**	∞	0,88	0,7	0,97	**3,06**	**99,66**
Letter										
Alatasetal	25,33	0,2	**0,99**	4,02	∞	0,75	0,34	0,23	7,17	51,65
MOPNAR	**60,6**	**0,29**	0,91	**6,19**	∞	**0,87**	**0,6**	**0,97**	**3,37**	**99,72**
Magic										
Alatasetal	11	**0,47**	**1**	1,26	956,59	**0,88**	0,34	0,91	4,73	89,9
MOPNAR	**99,4**	0,38	0,89	**8,21**	∞	0,86	**0,66**	**0,99**	**2,6**	**99,98**
Movement Libras										
Alatasetal	0	-	-	-	-	-	-	-	-	-
MOPNAR	**68,8**	**0,21**	**0,96**	**21,96**	∞	**0,96**	**0,9**	**1**	**2,67**	**95,33**
Optdigits										
Alatasetal	0	-	-	-	-	-	-	-	-	-
MOPNAR	**64,2**	**0,24**	**0,82**	**5,91**	∞	**0,77**	**0,56**	**0,95**	**3,51**	**100**
Penbased										
Alatasetal	0	-	-	-	-	-	-	-	-	-
MOPNAR	**78**	**0,29**	**0,92**	**6,8**	**Inf**	**0,89**	**0,72**	**0,99**	**3,03**	**99,86**
Pollution										
Alatasetal	17,2	**0,59**	**1**	6,86	∞	0,45	0,39	0,45	3,29	59,67
MOPNAR	**60,6**	0,21	0,98	**32,33**	∞	**0,97**	**0,85**	**1**	**2,36**	**96,67**
Quake										
Alatasetal	4,25	**0,67**	**1**	1,01	∞	0,1	0	0	**2,08**	98,06
MOPNAR	**43,6**	0,31	0,91	**8,63**	∞	**0,85**	**0,53**	**0,94**	2,31	**99,83**
Satimage										
Alatasetal	0	-	-	-	-	-	-	-	-	-
MOPNAR	**138,6**	**0,32**	**0,95**	**7,4**	∞	**0,93**	**0,8**	**0,99**	**3,79**	**100**
Segment										
Alatasetal	47	**0,53**	0,94	1,03	∞	0,26	0,02	0,21	4,15	**100**
MOPNAR	**88**	0,3	**0,98**	**15,44**	∞	**0,97**	**0,86**	**1**	**2,77**	99,98
Sonar										
Alatasetal	0	-	-	-	-	-	-	-	-	-
MOPNAR	**50,6**	**0,35**	**0,93**	**8,51**	∞	**0,89**	**0,65**	**0,98**	**2,73**	**99,04**
Spambase										
Alatasetal	0	-	-	-	-	-	-	-	-	-
MOPNAR	**74,4**	**0,33**	**0,8**	**6,72**	∞	**0,73**	**0,51**	**0,94**	**4,04**	**100**
Spectfheart										
Alatasetal	0	-	-	-	-	-	-	-	-	-
MOPNAR	**53,8**	**0,38**	**0,9**	**11,77**	∞	**0,83**	**0,6**	**0,94**	**2,92**	**98,96**
Stock										
Alatasetal	10,8	0,08	**1**	**96,75**	∞	0,91	0,72	0,73	**2,73**	21,04
MOPNAR	**78,8**	**0,25**	0,93	11,29	∞	**0,92**	**0,83**	**1**	2,92	**100**
Stulong										
Alatasetal	7,33	**0,72**	**0,99**	1,5	∞	0,24	0,08	0,11	2,95	99,25
MOPNAR	**74**	0,31	0,82	**3,87**	∞	**0,73**	**0,5**	**0,91**	**2,67**	**99,99**
Texture										
Alatasetal	0	-	-	-	-	-	-	-	-	-
MOPNAR	**95,6**	**0,3**	**0,94**	**11,18**	∞	**0,92**	**0,84**	**1**	**3,08**	**99,78**
Thyroid										
Alatasetal	**80**	0,32	**0,96**	1,02	∞	0,5	0,06	0,24	17,05	89,27
MOPNAR	57,6	**0,36**	0,9	**11,26**	∞	**0,83**	**0,55**	**0,94**	**3,38**	**99,99**
Vehicle										
Alatasetal	22,4	0,01	**1**	**77,37**	∞	**1**	0,79	0,61	4,27	0,24
MOPNAR	**78,6**	**0,29**	0,97	12,7	∞	0,97	**0,9**	**1**	**2,6**	**99,95**
Vowel										
Alatasetal	**89,6**	0,13	**1**	**63,92**	∞	0,76	0,17	0,64	8,33	92,63
MOPNAR	47,8	**0,17**	0,87	12,82	∞	**0,84**	**0,71**	**0,98**	**3,14**	**99,11**
Wdbc										
Alatasetal	0	-	-	-	-	-	-	-	-	-
MOPNAR	**71,8**	**0,32**	**0,97**	**13,08**	∞	**0,96**	**0,87**	**1**	**2,6**	**99,51**
Wine										
Alatasetal	28,2	**0,27**	**1**	**25,89**	∞	0,79	0,45	0,76	4,3	40,45
MOPNAR	**54**	0,26	0,94	16,97	∞	**0,92**	**0,78**	**0,99**	**2,79**	**100**

Tabla C.8: Resultados para las BDs Balance, Basketball, Bolts, Coil2000 y House16H en la comparación entre los algoritmos evolutivos y MOPNAR

Algoritmos	*#R*	Med_{Sop}	Med_{Conf}	Med_{Lift}	Med_{Conv}	Med_{FC}	$Med_{Netconf}$	$Med_{Yule'sQ}$	Med_{Amp}	*%Ejem*
					Balance					
EARMGA	**100**	0,49	**1**	1	1	0	0	0	**2**	**100**
GAR	0	-	-	-	-	-	-	-	-	-
GENAR	30	0,13	0,94	**2,05**	30,95	**0,89**	**0,56**	**0,92**	5	85,13
ARMMGA	1	0,28	0,68	1,47	1,67	0,4	0,36	0,64	**2**	27,04
MODENAR	38,4	0,13	0,64	1,38	∞	0,32	0,18	0,39	3,04	75,43
MOEA_Ghosh	24,6	**0,91**	0,95	1	1	0	0	0	4,32	**100**
MOPNAR	71	0,15	0,83	2,01	∞	0,71	0,43	0,81	2,95	**100**
					Basketball					
EARMGA	**100**	0,24	**1**	1,05	∞	0,18	0,03	0,12	**2**	**100**
GAR	2	0,77	0,89	1,02	1,13	0,11	0,11	0,37	**2**	97,09
GENAR	30	0,3	0,97	1,1	∞	0,68	0,12	0,64	5	89,8
ARMMGA	1	0,23	0,85	1,73	3,32	0,7	0,49	0,82	**2**	22,92
MODENAR	61,4	0,32	0,79	2,37	∞	0,33	0,15	0,29	2,3	97,5
MOEA_Ghosh	26,2	**0,86**	0,98	3,7	∞	0,18	0,12	0,24	3,71	**100**
MOPNAR	91,6	0,16	0,95	**47,73**	∞	**0,92**	**0,78**	**0,98**	2,33	99,79
					Bolts					
EARMGA	**100**	0,35	**1**	1,1	∞	0,2	0,07	0,2	**2**	**100**
GAR	33,2	0,21	0,98	4,21	∞	0,96	0,86	0,99	3,36	91,5
GENAR	30	0,14	**1**	1,57	∞	**1**	0,42	**1**	8	39
ARMMGA	1	0,46	**1**	1,3	∞	**1**	0,42	**1**	**2**	46
MODENAR	39,6	0,39	0,94	2,27	∞	0,48	0,41	0,54	3,93	68,5
MOEA_Ghosh	11,8	**0,76**	0,95	4,08	∞	0,33	0,29	0,37	6,33	**100**
MOPNAR	52	0,36	**1**	**14,16**	∞	0,99	**0,93**	**1**	2,34	**100**
					Coil2000					
EARMGA	72,4	0,33	**1**	1,01	∞	0,11	0,01	0,01	**2**	**100**
GAR	**197,6**	0,94	0,98	1,01	∞	0,06	0,03	0,08	2,07	**100**
GENAR	30	0,01	0,96	1,02	∞	0,37	0,02	0	86	13,74
ARMMGA	1	**1**	**1**	1	∞	0,39	0,07	0	**2**	99,78
MODENAR	35,6	0,05	0,69	510,26	∞	0,52	0,26	0,47	53,07	25,88
MOEA_Ghosh	49,6	0,01	0,33	**725,26**	∞	0,33	0,18	0,32	72,08	0,28
MOPNAR	28,4	0,36	0,88	5,41	∞	**0,84**	**0,67**	**0,96**	3,17	99,72
					House16H					
EARMGA	60,2	0,17	**1**	1,01	∞	0,28	0,01	0,01	3	98,56
GAR	**105,6**	0,76	0,9	1,03	1,33	0,2	0,17	0,48	**2,01**	**99,99**
GENAR	30	0,44	0,99	1,02	2,09	0,44	0,03	0,41	17	87,31
ARMMGA	1	**0,98**	**1**	1,01	1,13	0,09	0,04	0,16	3	97,97
MODENAR	64,4	0,69	0,99	1,15	∞	0,72	0,19	0,74	7	81
MOEA_Ghosh	19,8	0,6	0,79	**269,17**	∞	0,36	0,11	0,3	7,75	98,87
MOPNAR	91	0,32	0,88	6,08	∞	**0,84**	**0,65**	**0,97**	2,99	99,94

Tabla C.9: Resultados para las BDs Ionosphere, Letter, Magic, Movement Libras, Optdigits, Penbased y Pollution en la comparación entre los algoritmos evolutivos y MOPNAR

Algoritmos	*#R*	Med_{Sop}	Med_{Conf}	Med_{Lift}	Med_{Conv}	Med_{FC}	$Med_{Netconf}$	$Med_{Yule'sQ}$	Med_{Amp}	*%Ejem*
Ionosphere										
EARMGA	100	0,4	**1**	1,01	∞	0,01	0	0,01	**2**	**100**
GAR	39	0,21	0,93	1,62	∞	0,8	0,42	0,83	**2**	77,33
GENAR	30	0,3	0,99	1,55	∞	**0,97**	0,5	**0,98**	34	35,22
ARMMGA	1	**0,54**	0,87	1,41	3,02	0,65	0,64	0,92	**2**	53
MODENAR	28	0,26	0,76	**42,04**	∞	0,59	0,65	0,83	22,09	55,28
MOEA_Ghosh	**292**	0,01	0,07	21,08	∞	0,06	0,06	0,13	31,59	1,26
MOPNAR	72,8	0,3	0,93	13,91	∞	0,88	**0,7**	0,97	3,06	99,66
Letter										
EARMGA	**100**	0,35	**1**	1	∞	0,01	0	0	**2**	**100**
GAR	12,20	**0,58**	0,85	1,19	1,70	0,34	0,29	0,56	2,02	99,79
GENAR	30	0,02	0,32	8,15	2,36	0,29	**0,30**	**0,84**	17	67,99
ARMMGA	1	0,49	0,82	1,13	1,65	0,30	0,21	0,42	**2**	49,28
MODENAR	56,60	0,33	0,95	1,44	∞	0,67	0,06	0,58	7,20	90,46
MOEA_Ghosh	29	0,48	0,82	**58,52**	∞	0,35	0,20	0,37	9,60	97,33
MOPNAR	60,60	0,29	0,91	6,19	∞	**0,87**	**0,60**	**0,97**	3,37	99,72
Magic										
EARMGA	92	0,31	**1**	1,05	∞	0,05	0,02	0,03	**2**	**100**
GAR	57,8	0,67	0,91	1,1	2,33	0,48	0,35	0,74	2,08	96,84
GENAR	30	0,43	0,82	1,26	1,9	0,47	0,35	0,67	11	62,56
ARMMGA	1	**0,87**	0,96	1,05	9,08	0,47	0,41	0,89	**2**	86,31
MODENAR	73,6	0,46	0,94	**234,06**	∞	0,5	0,05	0,48	3,8	80,31
MOEA_Ghosh	29,4	0,72	0,9	1,46	∞	0,52	0,17	0,5	5,26	98,33
MOPNAR	**99,4**	0,38	0,89	8,21	∞	**0,86**	**0,66**	**0,99**	2,6	99,98
Movement Libras										
EARMGA	**100**	0,39	**1**	1	∞	0	0	0	**2**	**100**
GAR	2,6	**0,42**	0,94	3,73	12,62	0,9	**0,9**	**1**	**2**	53,28
GENAR	30	0,04	0,89	13,35	∞	0,89	0,86	0,99	91	53,73
ARMMGA	1	0,26	0,87	3,27	28,86	0,79	0,79	0,93	**2**	25,95
MODENAR	23,4	0,01	0,15	32,31	∞	0,04	0,15	0,17	61,61	3,17
MOEA_Ghosh	10,8	0,01	0,22	**79,47**	∞	0,22	0,22	0,22	80,08	0,28
MOPNAR	68,8	0,21	0,96	21,96	∞	**0,96**	**0,9**	**1**	2,67	95,33
Optdigits										
EARMGA	**89**	0,41	**1**	1,01	∞	0,04	0,01	0	**2**	**100**
GAR	65,8	0,71	0,97	1,02	∞	0,18	0,03	0,04	2,01	**100**
GENAR	16,2	0,01	**1**	10,1	∞	**1**	**0,91**	0,86	65	1,78
ARMMGA	1	**1**	**1**	1	1	0	0	0	**2**	99,98
MODENAR	8	0,01	0,69	**3668,12**	∞	0,52	0,01	0,02	44,48	0,2
MOEA_Ghosh	47,8	0,01	0,16	859,53	∞	0,16	0,01	0	54,8	0,05
MOPNAR	64,2	0,24	0,82	5,91	∞	0,77	0,56	**0,95**	3,51	**100**
Penbased										
EARMGA	**100**	0,36	**1**	1	1	0	0	0	**2**	**100**
GAR	2,00	**0,72**	0,87	1,04	1,22	0,18	0,18	0,48	**2**	94,95
GENAR	30	0,05	0,97	9,60	∞	**0,97**	**0,91**	**1**	17	44,79
ARMMGA	1	0,42	0,80	1,36	3	0,50	0,41	0,71	2,00	41,40
MODENAR	68,20	0,25	0,95	1,50	∞	0,84	0,33	0,86	7,20	50,04
MOEA_Ghosh	24,40	0,57	0,75	**62,26**	∞	0,27	0,14	0,44	9,18	96,50
MOPNAR	78	0,29	0,92	6,80	∞	0,89	0,72	0,99	3,03	99,86
Pollution										
EARMGA	**100**	0,25	**1**	1,19	∞	0,31	0,06	0,27	**2**	**100**
GAR	54	**0,67**	0,91	1,17	∞	0,53	0,43	0,77	2,03	**100**
GENAR	30	0,22	**1**	1,23	∞	0,98	0,24	0,98	16	48
ARMMGA	1	0,64	**1**	1,04	∞	**1**	0,1	**1**	**2**	63,34
MODENAR	34,4	0,27	0,91	2,94	∞	0,85	0,52	0,94	7,2	48,34
MOEA_Ghosh	26,8	0,18	0,67	8,89	∞	0,61	0,62	0,95	13,18	39
MOPNAR	60,6	0,21	0,98	**32,33**	∞	0,97	**0,85**	**1**	2,36	96,67

Tabla C.10: Resultados para las BDs Quake, Satimage, Segment, Sonar, Spambase, Spectfheart en la comparación entre los algoritmos evolutivos y MOPNAR

Algoritmos	*#R*	Med_{Sop}	Med_{Conf}	Med_{Lift}	Med_{Conv}	Med_{FC}	$Med_{Netconf}$	$Med_{Yule'sQ}$	Med_{Amp}	*%Ejem*
					Quake					
EARMGA	**100**	0,27	**1**	1	∞	0,01	0	0	**2**	**100**
GAR	1	0,44	0,84	0,98	0,88	-0,03	-0,05	-0,17	**2**	52,89
GENAR	30	0,55	0,95	1,01	1,09	0,09	0,02	0,1	4	81,78
ARMMGA	1	0,66	0,73	1,01	1,02	0,02	0,04	0,09	**2**	65,94
MODENAR	63,6	0,36	0,84	**117,09**	∞	0,31	0,09	0,22	2,09	92,84
MOEA_Ghosh	8	**0,86**	**1**	1,01	∞	0,18	0,01	0,14	3,09	**100**
MOPNAR	43,6	0,31	0,91	8,63	∞	**0,85**	**0,53**	**0,94**	2,31	99,83
					Satimage					
EARMGA	88	0,34	**1**	1,06	∞	0,03	0,02	0,02	**2**	**100**
GAR	207,8	0,91	0,97	1,04	2,28	0,4	0,38	0,77	2,13	**100**
GENAR	30	0,22	0,32	1,46	1,18	0,11	0,3	0,96	37	99,98
ARMMGA	1	**0,96**	0,99	1,02	2,25	0,56	0,54	0,98	**2**	95,44
MODENAR	48,6	0,53	0,98	1,25	∞	**0,93**	0,34	0,94	18,6	88,65
MOEA_Ghosh	**708**	0,01	0,01	**23,37**	∞	0,01	0,01	0	36,9	0,06
MOPNAR	138,6	0,32	0,95	7,4	∞	**0,93**	**0,8**	**0,99**	3,79	**100**
					Segment					
EARMGA	**99,2**	0,45	**1**	1,04	∞	0,08	0,02	0,04	**2**	**100**
GAR	18,8	0,36	0,89	2,49	3,61	0,58	0,47	0,73	**2**	97,97
GENAR	30	0,07	0,78	5,43	∞	0,74	0,7	0,93	20	83,49
ARMMGA	1	**0,92**	**1**	1,14	∞	0,2	0,2	0,2	**2**	91,33
MODENAR	58,8	0,33	0,97	1,72	∞	0,93	0,58	0,96	10,6	56,49
MOEA_Ghosh	28,2	0,36	0,86	**108,6**	∞	0,73	0,5	0,8	12,63	72,95
MOPNAR	88	0,3	0,98	15,44	∞	**0,97**	**0,86**	**1**	2,77	99,98
					Sonar					
EARMGA	**100**	0,27	**1**	1,52	∞	0,13	0,02	0,08	**2**	**100**
GAR	9,6	0,41	0,84	1,33	2,16	0,46	0,31	0,62	**2**	93,27
GENAR	30	0,04	0,92	1,77	∞	0,83	0,42	0,83	61	30,1
ARMMGA	1	**0,72**	0,91	1,05	1,61	0,28	0,22	0,56	**2**	72,02
MODENAR	14,8	0,01	0,25	52,63	∞	0,2	0,25	0,25	41,24	1,74
MOEA_Ghosh	26,4	0,01	0,35	**72,01**	∞	0,35	0,35	0,35	54,69	0,49
MOPNAR	50,6	0,35	0,93	8,51	∞	**0,89**	**0,65**	**0,98**	2,73	99,04
					Spambase					
EARMGA	54,6	0,27	**1**	1,01	∞	0,39	0,01	0	**2**	80,85
GAR	1,5	0,22	0,94	1,75	7,69	**0,84**	0,44	0,88	**2**	27,4
GENAR	30	0,44	0,62	1,04	1,05	0,05	0,06	0,12	58	84,8
ARMMGA	1	**1**	**1**	1	1	-0,01	-0,01	0	**2**	99,81
MODENAR	54	0,66	0,94	**128,4**	∞	0,39	0,1	0,5	36,4	98,83
MOEA_Ghosh	**81**	0,17	0,88	81,67	∞	0,81	**0,61**	0,85	37,47	31,51
MOPNAR	74,4	0,33	0,8	6,72	∞	0,73	0,51	**0,94**	4,04	**100**
					Spectfheart					
EARMGA	100	0,36	**1**	1,01	∞	0,02	0,01	0,02	**2**	**100**
GAR	30	0,70	0,88	1,08	1,63	0,33	0,30	0,65	**2**	99,85
GENAR	30	0,25	0,68	0,91	0,68	-0,13	-0,17	-0,45	45	60,15
ARMMGA	1	**0,74**	0,94	1,12	2,64	0,55	0,40	0,84	**2**	73,64
MODENAR	47	0,21	0,89	1,78	∞	0,77	0,48	0,86	22,60	50,79
MOEA_Ghosh	**458,60**	0,01	0,05	**11,83**	∞	0,05	0,05	0,07	43,65	0,60
MOPNAR	53,80	0,38	0,90	11,77	∞	**0,83**	**0,60**	**0,94**	2,92	98,96

Tabla C.11: Resultados para las BDs Stock, Stulong, Texture, Thyroid, Vehicle, Vowel, Wdbc y Wine en la comparación entre los algoritmos evolutivos y MOPNAR

Algoritmos	*#R*	Med_{Sop}	Med_{Conf}	Med_{Lift}	Med_{Conv}	Med_{FC}	$Med_{Netconf}$	$Med_{Yule'sQ}$	Med_{Amp}	*%Ejem*
					Stock					
EARMGA	**100**	0,37	**1**	1,01	∞	0,02	0,01	0,02	**2**	**100**
GAR	2	0,56	0,87	1,35	3,19	0,62	0,62	0,88	**2**	73,3
GENAR	30	0,29	0,92	1,69	∞	0,81	0,54	0,89	10	88,51
ARMMGA	1	0,37	0,77	1,63	2,25	0,56	0,56	0,86	**2**	36,22
MODENAR	63,8	0,48	0,92	1,75	∞	0,61	0,3	0,54	3	81,86
MOEA_Ghosh	19,8	**0,61**	0,91	**42,56**	∞	0,53	0,36	0,68	5,28	96,4
MOPNAR	78,8	0,25	0,93	11,29	∞	**0,92**	**0,83**	**1**	2,92	**100**
					Stulong					
EARMGA	92,6	0,27	**1**	1,01	∞	0,13	0,01	0,02	**2**	**100**
GAR	**157,4**	0,78	0,94	1,03	1,63	0,31	0,21	0,63	2,96	99,94
GENAR	30	**0,88**	0,99	1,01	1,02	0,02	0,01	0,04	5	95,26
ARMMGA	1	0,87	0,87	1,01	1,03	0,03	**0,62**	**0,91**	**2**	86,38
MODENAR	63,2	0,52	0,88	**13,94**	∞	0,27	0,06	0,27	3	99,28
MOEA_Ghosh	19,6	0,83	0,99	1,04	∞	0,51	0,23	0,6	3,47	99,92
MOPNAR	74	0,31	0,82	3,87	∞	**0,73**	0,5	**0,91**	2,67	99,99
					Texture					
EARMGA	**100**	0,27	**1**	3,52	∞	0,08	0,05	0,07	**2**	**100**
GAR	38,8	**0,72**	0,94	1,23	∞	0,72	0,67	0,92	2,03	97,98
GENAR	30	0,09	0,69	7,51	∞	0,66	0,68	0,99	41	98,18
ARMMGA	1	0,71	0,95	1,34	∞	0,86	**0,87**	0,98	**2**	70,42
MODENAR	29	0,07	0,48	4,42	∞	0,21	0,39	0,66	27,55	39,74
MOEA_Ghosh	81,4	0,01	0,17	**916,93**	∞	0,17	0,01	0	37,01	0,04
MOPNAR	95,6	0,3	0,94	11,18	∞	**0,92**	0,84	**1**	3,08	99,78
					Thyroid					
EARMGA	80,20	0,56	**1**	1,01	∞	0,03	0,01	0	**2**	**100**
GAR	**191,20**	0,83	0,92	1,02	1,17	0,14	0,13	0,46	2,01	99,97
GENAR	30	0,59	0,93	1	1,01	0,03	0	-0,01	22	94,30
ARMMGA	1	**1**	**1**	1	1	-0,01	-0,01	0	**2**	99,08
MODENAR	71,40	0,72	0,99	1,01	∞	0,23	0,02	0,16	12,80	95,89
MOEA_Ghosh	28,80	0,99	0,99	1	1	0	0	0,02	15,45	**100**
MOPNAR	57,60	0,36	0,90	**11,26**	∞	**0,83**	**0,55**	**0,94**	3,38	99,99
					Vehicle					
EARMGA	100	0,32	**1**	1,08	∞	0,11	0,04	0,09	**2**	**100**
GAR	24,80	0,67	0,94	1,31	∞	0,48	0,36	0,79	2,03	**100**
GENAR	30	0,09	0,67	2,62	2,60	0,56	0,48	0,76	19	66,39
ARMMGA	1,20	**0,74**	0,96	1,16	9,25	0,65	0,22	0,76	**2**	78,35
MODENAR	60,20	0,38	0,95	1,71	∞	0,88	0,61	0,95	8,60	65,16
MOEA_Ghosh	**470**	0,17	0,53	**50,16**	∞	0,49	0,39	0,73	15,92	31,07
MOPNAR	78,60	0,29	0,97	12,70	∞	**0,97**	**0,90**	**1**	2,60	99,95
					Vowel					
EARMGA	**100**	0,34	**1**	1,01	∞	0,01	0,01	0,01	**2**	**100**
GAR	2	**0,77**	0,88	1,02	1,12	0,1	0,1	0,35	**2**	97,04
GENAR	30	0,02	0,54	5,93	∞	0,5	0,47	0,85	14	63,26
ARMMGA	1	0,26	0,95	1,99	∞	**0,91**	0,64	0,94	**2**	25,34
MODENAR	63,4	0,17	0,66	3,08	∞	0,46	0,4	0,58	5	29,6
MOEA_Ghosh	21	0,62	0,81	**56,42**	∞	0,24	0,12	0,33	6,53	99,94
MOPNAR	47,8	0,17	0,87	12,82	∞	0,84	**0,71**	**0,98**	3,14	99,11
					Wdbc					
EARMGA	100	0,25	**1**	1,52	∞	0,17	0,06	0,1	**2**	**100**
GAR	88,8	0,54	0,86	1,19	2,49	0,37	0,32	0,49	**2**	98,84
GENAR	30	0,44	0,94	1,53	6,47	0,84	0,6	0,94	31	71,36
ARMMGA	1	**0,94**	0,98	1,02	∞	0,46	0,21	0,73	**2**	93,47
MODENAR	37,8	0,19	0,77	**35,57**	∞	0,6	0,36	0,71	17,73	67,91
MOEA_Ghosh	**1338,2**	0,01	0,01	2,01	∞	0,01	0,01	0,01	30,9	0,57
MOPNAR	71,8	0,32	0,97	13,08	∞	**0,96**	**0,87**	**1**	2,6	99,51
					Wine					
EARMGA	**100**	0,38	**1**	1,02	∞	0,05	0,01	0,05	**2**	**100**
GAR	8,2	0,22	0,98	2,79	∞	0,95	0,76	0,97	**2**	55,96
GENAR	30	0,19	**1**	3,02	∞	**1**	**0,82**	**1**	14	68,66
ARMMGA	1	0,26	0,85	2,7	8,53	0,75	0,72	0,92	**2**	25,06
MODENAR	54,4	0,39	0,91	6,07	∞	0,63	0,33	0,72	5,4	71,58
MOEA_Ghosh	22,4	**0,44**	0,85	10,37	∞	0,55	0,42	0,68	8,54	80,57
MOPNAR	54	0,26	0,94	**16,97**	∞	0,92	0,78	0,99	2,79	**100**

Tabla C.12: Resultados para las BDs Balance, Basketball, Bolts, Coil2000, House16H, Ionosphere, Letter, Magic, Movement Libras, Optdigits, Penbased, Pollution y Quake en la comparación entre los algoritmos clásicos y MOPNAR

Algoritmos	*#R*	Med_{Sop}	Med_{Conf}	Med_{Lift}	Med_{Conv}	Med_{FC}	$Med_{Netconf}$	$Med_{Yule'sQ}$	Med_{Amp}	*%Ejem*
Balance										
Apriori	2	0,14	**0,86**	**6,25**	5,77	**0,83**	**0,86**	**1**	3	27,20
Eclat	2	0,14	**0,86**	**6,25**	5,77	**0,83**	**0,86**	**1**	3	27,20
NSGA-II	69,80	**0,17**	0,83	2,17	∞	0,71	0,47	0,82	3,32	88,04
MOPNAR	**71**	0,15	0,83	2,01	∞	0,71	0,43	0,81	**2,95**	**100**
Basketball										
Apriori	4	0,15	0,87	4,88	6,65	0,84	0,81	**0,99**	2,75	33,34
Eclat	4	0,15	0,87	4,88	6,65	0,84	0,81	**0,99**	2,75	33,34
NSGA-II	32,80	**0,23**	0,92	28,76	∞	0,87	0,68	0,95	**2**	81,46
MOPNAR	**91,60**	0,16	**0,95**	**47,73**	∞	**0,92**	**0,78**	0,98	2,33	**99,79**
Bolts										
Apriori	**1246**	0,15	0,99	7,16	∞	0,98	**0,96**	**1**	**4,36**	97,50
Eclat	**1246**	0,15	0,99	7,16	∞	0,98	**0,96**	**1**	**4,36**	97,50
NSGA-II	97,40	0,21	**1**	5,79	∞	**1**	0,95	**1**	2,76	94,50
MOPNAR	52,00	**0,36**	**1**	**14,16**	∞	0,99	0,93	**1**	2,34	**100**
Coil2000										
Apriori	-	-	-	-	-	-	-	-	-	-
Eclat	-	-	-	-	-	-	-	-	-	-
NSGA-II	**81,60**	0,23	**0,92**	**60,49**	∞	**0,87**	0,63	0,93	5	99,70
MOPNAR	28,40	**0,36**	0,88	5,41	∞	0,84	**0,67**	**0,96**	**3,17**	**99,72**
House16H										
Apriori	**1749917**	0,22	**0,97**	2,19	∞	0,83	0,45	0,88	8,65	**100**
Eclat	**1749917**	0,22	**0,97**	2,19	∞	0,83	0,45	0,88	8,65	**100**
NSGA-II	99,20	**0,36**	0,95	**389,34**	∞	**0,93**	0,61	0,90	4,05	97,78
MOPNAR	91,00	0,32	0,88	6,08	∞	0,84	**0,65**	**0,97**	**2,99**	99,94
Ionosphere										
Apriori	**2,1e+08**	0,12	**0,98**	4,40	∞	**0,97**	0,72	**1**	10,79	**100**
Eclat	**2,1e+08**	0,12	**0,98**	4,40	∞	**0,97**	0,72	**1**	10,79	**100**
NSGA-II	99,60	0,25	0,93	**14,34**	∞	0,89	**0,75**	0,98	3,28	82,57
MOPNAR	72,80	**0,30**	0,93	13,91	∞	0,88	0,70	0,97	**3,06**	99,66
Letter										
Apriori	**3916**	0,14	0,88	4,66	∞	0,81	**0,73**	0,93	4,55	99,49
Eclat	**3916**	0,14	0,88	4,66	∞	0,81	**0,73**	0,93	4,55	99,49
NSGA-II	82,2	0,23	0,9	**58,62**	∞	0,85	0,61	0,92	3,77	86,71
MOPNAR	60,6	**0,29**	**0,91**	6,19	∞	**0,87**	0,6	**0,97**	**3,37**	**99,72**
Magic										
Apriori	**9785**	0,19	**0,96**	2,73	∞	0,87	0,52	0,95	5,53	99,96
Eclat	**9785**	0,19	**0,96**	2,73	∞	0,87	0,52	0,95	5,53	99,96
NSGA-II	100,00	**0,45**	**0,96**	**325,70**	∞	**0,94**	0,62	0,93	3,41	99,08
MOPNAR	99,40	0,38	0,89	8,21	∞	0,86	**0,66**	**0,99**	**2,60**	**99,98**
Movement Libras										
Apriori	-	-	-	-	-	-	-	-	-	-
Eclat	-	-	-	-	-	-	-	-	-	-
NSGA-II	47	0,18	0,95	**66,42**	∞	0,94	0,88	**1**	**2,26**	78,95
MOPNAR	**68,80**	**0,21**	**0,96**	21,96	∞	**0,96**	**0,90**	**1**	2,67	**95,33**
Optdigits										
Apriori	-	-	-	-	-	-	-	-	-	-
Eclat	-	-	-	-	-	-	-	-	-	-
NSGA-II	**94,20**	**0,25**	**0,91**	**67,77**	∞	**0,87**	**0,57**	0,90	3,73	95,69
MOPNAR	64,20	0,24	0,82	5,91	∞	0,77	0,56	**0,95**	**3,51**	**100**
Penbased										
Apriori	**1918**	0,14	**0,92**	4,23	∞	0,82	0,63	0,87	4,07	99,5
Eclat	**1918**	0,14	**0,92**	4,23	∞	0,82	0,63	0,87	4,07	99,5
NSGA-II	100	0,22	0,91	**15,38**	∞	0,88	0,67	0,93	3,5	80,31
MOPNAR	78	**0,29**	0,92	6,8	∞	**0,89**	**0,72**	**0,99**	**3,03**	**99,86**
Pollution										
Apriori	**41510**	0,13	0,95	5,84	∞	0,93	**0,86**	0,99	5,88	**100**
Eclat	**41510**	0,13	0,95	5,84	∞	0,93	**0,86**	0,99	5,88	**100**
NSGA-II	29,00	**0,28**	0,95	23,77	∞	0,93	0,84	0,99	**2**	89
MOPNAR	60,60	0,21	**0,98**	**32,33**	∞	**0,97**	0,85	**1**	2,36	96,67
Quake										
Apriori	18	0,25	0,91	1	1,15	0,11	-0,01	0,51	2,56	90,55
Eclat	18	0,25	0,91	1	1,15	0,11	-0,01	0,51	2,56	90,55
NSGA-II	**88,40**	0,28	**0,94**	**55,01**	∞	**0,89**	**0,60**	**0,94**	2,80	94,23
MOPNAR	43,60	**0,31**	0,91	8,63	∞	0,85	0,53	**0,94**	**2,31**	**99,83**

Tabla C.13: Resultados para las BDs Satimage, Segment, Sonar, Spambase, Stock, Stulong, Texture, Thyroid, Vehicle, Wine, Wdbc y Vowel en la comparación entre los algoritmos clásicos y MOPNAR

Algoritmos	*#R*	Med_{Sop}	Med_{Conf}	Med_{Lift}	Med_{Conv}	Med_{FC}	$Med_{Netconf}$	$Med_{Yule'sQ}$	Med_{Amp}	*%Ejem*
Satimage										
Apriori	-	-	-	-	-	-	-	-	-	-
Eclat	-	-	-	-	-	-	-	-	-	-
NSGA-II	100	0,31	0,89	**25,08**	∞	0,87	0,67	0,93	**3,97**	99,95
MOPNAR	**138,60**	**0,32**	**0,95**	7,40	∞	**0,93**	**0,80**	**0,99**	3,79	**100**
Segment										
Apriori	-	-	-	-	-	-	-	-	-	-
Eclat	-	-	-	-	-	-	-	-	-	-
NSGA-II	**89**	0,28	0,97	**76,96**	∞	0,95	0,80	0,96	2,78	98,39
MOPNAR	88	**0,30**	**0,98**	15,44	∞	**0,97**	**0,86**	**1**	**2,77**	**99,98**
Sonar										
Apriori	-	-	-	-	-	-	-	-	-	-
Eclat	-	-	-	-	-	-	-	-	-	-
NSGA-II	**69,60**	0,26	0,90	**15,80**	∞	0,83	**0,68**	0,96	2,79	75
MOPNAR	50,60	**0,35**	**0,93**	8,51	∞	**0,89**	0,65	**0,98**	**2,73**	**99,04**
Spambase										
Apriori	-	-	-	-	-	-	-	-	-	-
Eclat	-	-	-	-	-	-	-	-	-	-
NSGA-II	**99,40**	**0,37**	**0,93**	**58,23**	∞	**0,86**	0,50	0,93	4,93	99,17
MOPNAR	74,40	0,33	0,80	6,72	∞	0,73	**0,51**	**0,94**	**4,04**	**100**
Spectfheart										
Apriori	-	-	-	-	-	-	-	-	-	-
Eclat	-	-	-	-	-	-	-	-	-	-
NSGA-II	**100**	0,31	**0,92**	6,75	∞	**0,86**	**0,61**	**0,95**	3,84	93,11
MOPNAR	53,8	**0,38**	0,9	**11,77**	∞	0,83	0,6	0,94	**2,92**	**98,96**
Stock										
Apriori	**855**	0,13	0,91	4,77	∞	0,88	0,76	0,98	4,16	99,48
Eclat	**855**	0,13	0,91	4,77	∞	0,88	0,76	0,98	4,16	99,48
NSGA-II	100	0,20	**0,93**	**30,03**	∞	**0,92**	**0,85**	**1**	3,07	82,17
MOPNAR	78,80	**0,25**	**0,93**	11,29	∞	**0,92**	0,83	**1**	**2,92**	**100**
Stulong										
Apriori	89	0,31	**0,93**	1,22	∞	0,43	0,14	0,64	3,26	99,86
Eclat	89	0,31	**0,93**	1,22	∞	0,43	0,14	0,64	3,26	99,86
NSGA-II	**97,20**	**0,36**	0,91	**29,11**	∞	**0,82**	**0,51**	**0,92**	3,28	99,10
MOPNAR	74	0,31	0,82	3,87	∞	0,73	0,50	0,91	**2,67**	**99,99**
Texture										
Apriori	-	-	-	-	-	-	-	-	-	-
Eclat	-	-	-	-	-	-	-	-	-	-
NSGA-II	**100**	0,26	**0,96**	**127,61**	∞	**0,94**	0,81	0,96	3,09	97,83
MOPNAR	95,60	**0,30**	0,94	11,18	∞	0,92	**0,84**	**1**	**3,08**	**99,78**
Thyroid										
Apriori	-	-	-	-	-	-	-	-	-	-
Eclat	-	-	-	-	-	-	-	-	-	-
NSGA-II	**100**	**0,42**	**0,95**	**117,31**	∞	**0,88**	0,47	0,91	3,55	97,99
MOPNAR	57,6	0,36	0,9	11,26	∞	0,83	**0,55**	**0,94**	**3,38**	**99,99**
Vehicle										
Apriori	**141107**	0,14	0,95	4,18	∞	0,91	0,76	0,95	6,4	**100**
Eclat	**141107**	0,14	0,95	4,18	∞	0,91	0,76	0,95	6,4	**100**
NSGA-II	60,8	0,26	0,96	**29,27**	∞	0,95	**0,92**	**1**	**2,42**	88,28
MOPNAR	78,6	**0,29**	**0,97**	12,7	∞	**0,97**	0,9	**1**	2,6	99,95
Wdbc										
Apriori	-	-	-	-	-	-	-	-	-	-
Eclat	-	-	-	-	-	-	-	-	-	-
NSGA-II	**95,80**	**0,32**	0,95	**22,95**	∞	0,94	0,84	**1**	2,91	92,45
MOPNAR	71,80	**0,32**	**0,97**	13,08	∞	**0,96**	**0,87**	**1**	**2,60**	**99,51**
Wine										
Apriori	**1348**	0,13	0,91	5,97	∞	0,87	**0,82**	0,98	4,07	**100**
Eclat	**1348**	0,13	0,91	5,97	∞	0,87	**0,82**	0,98	4,07	**100**
NSGA-II	36,80	0,18	0,87	**25,86**	∞	0,84	0,75	0,98	**2,28**	73,15
MOPNAR	54,00	**0,26**	**0,94**	16,97	∞	**0,92**	0,78	**0,99**	2,79	**100**
Vowel										
Apriori	**235**	0,14	**0,98**	2,97	∞	**0,96**	0,69	**0,98**	3,48	**100**
Eclat	**235**	0,14	**0,98**	2,97	∞	**0,96**	0,69	**0,98**	3,48	**100**
NSGA-II	53,80	0,14	0,83	**62,56**	∞	0,79	0,66	0,95	**2,53**	67,82
MOPNAR	47,80	**0,17**	0,87	12,82	∞	0,84	**0,71**	**0,98**	3,14	99,11

C.3. Resultados obtenidos para evaluar el método NIGAR

Tabla C.14: Resultados para las BDs Balance, Basketball, Bolts, Coil2000 y House16H en la comparación entre los algoritmos evolutivos y NIGAR

Algoritmos	*#R*	Med_{Sop}	Med_{Conf}	Med_{Lift}	Med_{Conv}	Med_{FC}	$Med_{Netconf}$	$Med_{Yule'sQ}$	Med_{Amp}	Med_{Div}	*%Ejem*
					Balance						
EARMGA	**100**	**0,5**	**1**	1	1	0	0	0	**2**	0,46	**100**
GENAR	30	0,13	0,94	2,04	31,04	**0,89**	0,55	**0,92**	5	0,6	85,64
GAR	0	-	-	-	-	-	-	-	-	-	-
Alatasetal	24,4	0,31	**1**	1,1	∞	0,06	0,03	0,05	4,35	0,48	**100**
NIGAR	26,4	0,06	0,92	**4,25**	∞	0,86	**0,69**	0,91	3,81	**0,82**	89,96
					Basketball						
EARMGA	**100**	0,27	**1**	1,02	∞	0,06	0,01	0,05	**2**	0,55	**100**
GENAR	30	0,3	0,97	1,12	∞	0,7	0,13	0,67	5	0,3	91,04
GAR	2	0,75	0,88	1,02	1,12	0,11	0,11	0,36	**2**	0	96,88
Alatasetal	8,6	**0,98**	**1**	1	1	-0,01	-0,01	0	3,2	0,09	**100**
NIGAR	49,8	0,04	0,99	**9,43**	∞	**0,98**	**0,8**	**0,99**	2,15	**0,95**	91,67
					Bolts						
EARMGA	**100**	0,34	**1**	1,05	∞	0,15	0,03	0,15	**2**	0,57	**100**
GENAR	30	0,13	**1**	1,62	∞	**1**	0,42	**1**	8	0,27	44
GAR	43	0,21	0,99	4,14	∞	0,97	0,88	**1**	3,31	0,44	81,5
Alatasetal	21	**0,95**	**1**	1,04	∞	0,14	0,14	0,14	3,64	0,02	95
NIGAR	9,8	0,3	0,98	**5,51**	∞	0,98	**0,93**	**1**	**2**	**0,82**	**100**
					Coil2000						
EARMGA	77	0,42	**1**	1,01	∞	0,05	0,01	0	**2**	0,55	**100**
GENAR	30	0,01	0,96	1,02	∞	0,34	0,02	0,01	86	0,53	14,48
GAR	**197**	**0,94**	0,97	1,01	∞	0,04	0,03	0,08	2,08	0,37	**100**
Alatasetal	0	-	-	-	-	-	-	-	-	-	-
NIGAR	25,4	0,28	0,93	**3,65**	∞	**0,89**	**0,86**	**0,98**	2,04	**0,82**	99,96
					House16H						
EARMGA	75,6	0,3	**1**	1	∞	0,05	0	0	2,2	0,54	99,95
GENAR	30	0,45	**1**	1,01	2,56	0,52	0,02	0,5	17	0,14	86,94
GAR	**112,8**	**0,77**	0,9	1,03	1,33	0,2	0,17	0,49	2,01	0,28	**99,98**
Alatasetal	91	0,19	0,99	1,03	∞	0,58	0,03	0,41	8,75	0,51	98,24
NIGAR	6	0,24	0,92	**3,67**	15	**0,88**	**0,83**	**0,98**	**2**	**0,86**	82,18

Tabla C.15: Resultados para las BDs Ionosphere, Magic, Movement Libras, Optdigits y Pollution en la comparación entre los algoritmos evolutivos y NIGAR

Algoritmos	*#R*	Med_{Sop}	Med_{Conf}	Med_{Lift}	Med_{Conv}	Med_{FC}	$Med_{Netconf}$	$Med_{Yule'sQ}$	Med_{Amp}	Med_{Div}	*%Ejem*
Ionosphere											
EARMGA	**100**	0,4	**1**	1	1	0	0	0	**2**	0,59	**100**
GENAR	30	0,3	0,99	1,55	∞	**0,97**	0,5	**0,98**	34	0,04	34,99
GAR	37,6	0,21	0,92	1,68	∞	0,81	0,44	0,84	**2**	0,44	81,26
Alatasetal	96	**0,79**	**1**	1,01	∞	0,43	0,03	0,49	9,69	0,05	99,44
NIGAR	24,8	0,2	0,85	**3,77**	∞	0,78	**0,74**	0,96	2,01	**0,88**	99,83
Letter											
EARMGA	**100**	0,36	**1**	1	∞	0	0	0	**2**	0,6	**100**
GENAR	30	0,02	0,27	**6,9**	1,92	0,24	0,25	0,82	17	0,63	72,76
GAR	13,4	**0,6**	0,85	1,15	1,55	0,29	0,25	0,49	**2**	0,37	99,95
Alatasetal	35,33	0,21	0,99	4,93	∞	0,76	0,22	0,36	7	0,42	54,28
NIGAR	18,8	0,15	0,89	5,94	∞	**0,84**	**0,76**	**0,96**	2,01	**0,87**	98,17
Magic											
EARMGA	**96**	0,33	**1**	1	∞	0,01	0	0	**2**	0,58	**100**
GENAR	30	0,43	0,81	1,25	1,85	0,46	0,34	0,66	11	0,04	62,81
GAR	64,2	**0,67**	0,91	1,11	2,32	0,49	0,35	0,74	2,11	0,18	97,47
Alatasetal	11	0,47	**1**	1,26	956,59	0,88	0,34	0,91	4,73	0,05	89,9
NIGAR	6,6	0,26	0,94	**3**	15,3	**0,91**	**0,85**	**0,99**	**2**	**0,85**	94,87
Movement Libras											
EARMGA	**100**	**0,4**	**1**	1	1	0	0	0	**2**	0,62	**100**
GENAR	30	0,03	0,91	**13,5**	∞	0,9	0,87	0,99	91	0,68	55,23
GAR	3	0,31	0,89	3,69	6,52	0,83	0,83	0,99	**2**	0,29	49,31
Alatasetal	0	-	-	-	-	-	-	-	-	-	-
NIGAR	27,8	0,27	0,98	3,64	∞	**0,97**	**0,95**	**1**	**2**	**0,86**	**100**
Optdigits											
EARMGA	**97,8**	0,41	**1**	1	∞	0,03	0	0	**2**	0,59	**100**
GENAR	17	0,01	**1**	**10,1**	∞	**1**	**0,91**	0,89	63	0,56	2,11
GAR	63,6	**0,71**	0,97	1,02	∞	0,17	0,04	0,1	2,01	0,49	**100**
Alatasetal	0	-	-	-	-	-	-	-	-	-	-
NIGAR	28,8	0,28	0,87	3,24	∞	0,75	0,71	**0,95**	2,05	**0,82**	**100**
Penbased											
EARMGA	**100**	0,42	**1**	1	1	0	0	0	**2**	0,58	**100**
GENAR	30	0,05	0,96	**9,56**	∞	**0,96**	**0,91**	**1**	17	0,62	46,29
GAR	2	**0,72**	0,87	1,04	1,22	0,18	0,18	0,48	**2**	0	94,95
Alatasetal	0	-	-	-	-	-	-	-	-	-	-
NIGAR	15	0,18	0,89	3,64	∞	0,83	0,73	0,95	2,06	**0,88**	98,64
Pollution											
EARMGA	**100**	0,21	**1**	1,08	∞	0,16	0,03	0,14	**2**	0,7	**100**
GENAR	30	0,23	**1**	1,23	∞	**0,99**	0,24	0,98	16	0,15	47,33
GAR	61,4	**0,67**	0,92	1,17	∞	0,56	0,46	0,78	**2**	0,19	**100**
Alatasetal	15,4	0,59	**1**	6,86	∞	0,43	0,39	0,43	3,32	0,01	59,67
NIGAR	57,8	0,08	0,98	**8,35**	∞	0,97	**0,84**	**1**	2,07	**0,93**	**100**

Tabla C.16: Resultados para las BDs Quake, Satimage, Segment, Sonar y Spambase en la comparación entre los algoritmos evolutivos y NIGAR

Algoritmos	*#R*	Med_{Sop}	Med_{Conf}	Med_{Lift}	Med_{Conv}	Med_{FC}	$Med_{Netconf}$	$Med_{Yule'sQ}$	Med_{Amp}	Med_{Div}	*%Ejem*
					Quake						
EARMGA	**100**	0,3	**1**	1	1	0	0	0	**2**	0,45	**100**
GENAR	30	0,55	0,95	1,01	1,09	0,09	0,01	0,1	4	0,07	81,89
GAR	1	0,45	0,85	0,99	0,91	-0,02	-0,03	-0,13	**2**	0	53,26
Alatasetal	4,25	**0,67**	**1**	1,01	∞	0,1	0	0	2,08	0,2	98,06
NIGAR	3,6	0,28	0,9	**2,51**	11,4	**0,81**	**0,72**	**0,95**	**2**	**0,75**	83,15
					Satimage						
EARMGA	88,8	0,38	**1**	1	∞	0,01	0	0	**2**	0,56	**100**
GENAR	30	0,22	0,31	1,42	∞	0,1	0,3	0,97	37	0,25	99,98
GAR	**206,6**	**0,91**	0,97	1,04	2,22	0,39	0,37	0,76	2,1	0,42	**100**
Alatasetal	0	-	-	-	-	-	-	-	-	-	-
NIGAR	24	0,24	0,92	**3,83**	12,27	**0,88**	**0,85**	**0,99**	**2**	**0,82**	93,94
					Segment						
EARMGA	**99,4**	0,39	**1**	1,05	∞	0,07	0,03	0,06	**2**	0,58	**100**
GENAR	30	0,08	0,77	**5,37**	∞	0,73	0,7	0,93	20	0,53	85,13
GAR	20,6	0,4	0,89	2,16	3,45	0,52	0,4	0,64	**2**	0,4	97,18
Alatasetal	63	**0,49**	0,96	1,04	∞	0,32	0,03	0,25	4,26	0,36	99,96
NIGAR	12,6	0,22	0,99	4,91	∞	**0,98**	**0,97**	**1**	**2**	**0,88**	99,76
					Sonar						
EARMGA	**100**	0,33	**1**	1,03	∞	0,03	0,01	0,02	**2**	0,62	**100**
GENAR	30	0,04	0,94	1,81	∞	**0,87**	0,44	0,88	61	0,47	30,96
GAR	7,2	**0,38**	0,83	1,3	2,05	0,45	0,29	0,59	**2**	0,21	69,62
Alatasetal	0	-	-	-	-	-	-	-	-	-	-
NIGAR	22,8	0,26	0,89	**7,56**	∞	0,81	**0,77**	**0,97**	2,01	**0,84**	**100**
					Spambase						
EARMGA	**61,2**	0,25	**1**	1,01	∞	0,35	0,01	0	**2**	0,63	81,46
GENAR	30	0,45	0,62	1,03	1,05	0,05	0,06	0,12	58	0,06	**86,21**
GAR	10,6	**0,55**	0,9	1,09	1,74	0,41	0,19	0,55	**2**	0,02	60,59
Alatasetal	0	-	-	-	-	-	-	-	-	-	-
NIGAR	11,2	0,12	0,95	**73,02**	∞	**0,9**	**0,87**	**0,98**	**2**	**0,84**	76,75
					Spectfheart						
EARMGA	**100**	0,36	**1**	1,01	∞	0,02	0,01	0,02	**2**	0,6	**100**
GENAR	30	0,25	0,68	0,91	0,68	-0,13	-0,17	-0,45	45	0,14	60,15
GAR	33,4	**0,7**	0,89	1,09	1,69	0,35	0,32	0,67	**2**	0,42	99,85
Alatasetal	0	-	-	-	-	-	-	-	-	-	-
NIGAR	72,2	0,13	0,98	**29,01**	∞	**0,92**	**0,92**	**1**	2,03	**0,89**	**100**

Tabla C.17: Resultados para las BDs Stock, Stulong, Texture, Vowel, Wdbc y Wine en la comparación entre los algoritmos evolutivos y NIGAR

Algoritmos	*#R*	Med_{Sop}	Med_{Conf}	Med_{Lift}	Med_{Conv}	Med_{FC}	$Med_{Netconf}$	$Med_{Yule'sQ}$	Med_{Amp}	Med_{Div}	*%Ejem*
					Stock						
EARMGA	**100**	0,38	**1**	1,01	∞	0,01	0,01	0,01	**2**	0,57	**100**
GENAR	30	0,3	0,92	1,64	∞	0,81	0,52	0,88	10	0,24	87,54
GAR	2	**0,52**	0,88	1,52	3,74	0,72	0,72	0,95	**2**	0	65,9
Alatasetal	7	0,11	0,99	**76,82**	∞	0,89	0,71	0,75	2,57	0,06	20,45
NIGAR	9,2	0,28	0,95	3,01	∞	**0,93**	**0,87**	**0,99**	2,07	**0,86**	99,41
					Stulong						
EARMGA	94	0,3	**1**	1,01	∞	0,06	0,01	0	**2**	0,5	**100**
GENAR	30	**0,89**	0,99	1,01	1,02	0,02	0,01	0,06	5	0,02	95,12
GAR	**161,6**	0,78	0,93	1,03	1,62	0,3	0,2	0,61	2,98	0,04	99,94
Alatasetal	7,67	0,72	**1**	1,48	∞	0,23	0,08	0,11	2,96	0,08	99,25
NIGAR	4,2	0,26	0,79	**16,82**	∞	**0,62**	**0,64**	**0,9**	2,04	**0,78**	84,99
					Texture						
EARMGA	**100**	0,24	**1**	1,17	∞	0,08	0,03	0,07	**2**	0,63	**100**
GENAR	30	0,09	0,68	**7,49**	∞	0,65	0,68	0,99	41	0,48	98,25
GAR	43,8	**0,71**	0,93	1,24	∞	0,69	0,66	0,93	2,01	0,38	97,85
Alatasetal	0	-	-	-	-	-	-	-	-	-	-
NIGAR	17,4	0,27	0,97	3,72	∞	**0,95**	**0,92**	**1**	**2**	**0,82**	95,85
					Thyroid						
EARMGA	88	0,57	**1**	1	∞	0,01	0	0	**2**	0,48	**100**
GENAR	30	0,69	0,93	1,01	1,05	0,05	0,02	0,07	22	0,04	95,5
GAR	**191,2**	**0,82**	0,92	1,02	1,17	0,13	0,13	0,45	2,01	0,03	99,91
Alatasetal	86,5	0,26	0,97	1,02	∞	0,58	0,05	0,22	16,75	0,43	89,92
NIGAR	6,2	0,23	0,93	**14,29**	∞	**0,8**	**0,85**	**0,95**	2,02	**0,79**	98,16
					Vehicle						
EARMGA	**100**	0,32	**1**	1,08	∞	0,11	0,04	0,09	**2**	0,61	**100**
GENAR	30	0,09	0,67	2,62	2,6	0,56	0,48	0,76	19	0,42	66,39
GAR	26,4	**0,67**	0,94	1,25	∞	0,42	0,3	0,74	2,02	0,21	**100**
Alatasetal	22,4	0,01	**1**	**77,37**	∞	**1**	0,79	0,61	4,27	0,16	0,24
NIGAR	12,8	0,27	0,97	3,46	∞	0,95	**0,92**	**1**	**2**	**0,86**	99,98
					Wdbc						
EARMGA	**100**	0,27	**1**	1,09	∞	0,06	0,03	0,05	**2**	0,61	**100**
GENAR	30	0,44	0,94	1,53	6,42	0,84	0,6	0,94	31	0,08	72,03
GAR	90,4	**0,54**	0,86	1,18	2,35	0,35	0,29	0,45	**2**	0,09	99,13
Alatasetal	0	-	-	-	-	-	-	-	-	-	-
NIGAR	13,6	0,27	0,96	**3,49**	∞	**0,93**	**0,9**	**0,99**	**2**	**0,84**	98,53
					Wine						
EARMGA	**100**	**0,4**	**1**	1,01	∞	0,03	0,01	0,02	**2**	0,58	**100**
GENAR	30	0,2	**1**	3,02	∞	**1**	0,83	**1**	14	0,28	66,41
GAR	7,6	0,21	0,98	2,83	∞	0,96	0,79	0,99	**2**	0,54	45,29
Alatasetal	27	0,27	**1**	**40,9**	∞	0,81	0,47	0,78	4,34	0,1	40,23
NIGAR	6,6	0,28	0,94	2,79	∞	0,9	**0,84**	0,99	**2**	**0,86**	94,61
					Vowel						
EARMGA	**100**	0,34	**1**	1	∞	0	0	0	**2**	0,62	**100**
GENAR	30	0,02	0,56	6,06	∞	0,51	0,48	0,86	14	0,7	63,64
GAR	1,67	**0,68**	0,86	1,04	1,23	0,18	0,17	0,44	**2**	0	86,74
Alatasetal	91,8	0,13	**1**	**72,54**	∞	0,76	0,17	0,62	8,36	0,56	94,47
NIGAR	8,4	0,27	0,95	5,81	∞	**0,92**	**0,89**	**0,95**	2,02	**0,85**	**100**

Tabla C.18: Resultados para las BDs Balance, Basketball, Bolts, Coil2000, House16H, Ionosphere, Magic, Movement Libras, Optdigits, Pollution, Quake y Satimage en la comparación entre los algoritmos clásicos y NIGAR

Algoritmos	*#R*	Med_{Sop}	Med_{Conf}	Med_{Lift}	Med_{Conv}	Med_{FC}	$Med_{Netconf}$	$Med_{Yule'sQ}$	Med_{Amp}	Med_{Div}	*%Ejem*
Balance											
Apriori	2	**0,14**	0,86	**6,25**	5,77	0,83	**0,86**	**1**	**3**	**1**	27,2
Eclat	2	**0,14**	0,86	**6,25**	5,77	0,83	**0,86**	**1**	**3**	**1**	27,2
Clearing	24,2	0,08	0,82	2,74	∞	0,68	0,49	0,73	3,47	0,73	89,48
NIGAR	**26,4**	0,06	**0,92**	4,25	∞	**0,86**	0,69	0,91	3,81	0,82	**89,96**
Basketball											
Apriori	4	**0,15**	0,87	4,88	6,65	0,84	0,81	**0,99**	2,75	0,5	33,34
Eclat	4	**0,15**	0,87	4,88	6,65	0,84	0,81	**0,99**	2,75	0,5	33,34
Clearing	**57**	0,03	0,98	**32,87**	∞	0,97	**0,91**	**0,99**	**2**	**0,99**	84,17
NIGAR	49,8	0,04	**0,99**	9,43	∞	**0,98**	0,8	**0,99**	2,15	0,95	**91,67**
Bolts											
Apriori	**1246**	0,15	**0,99**	7,16	∞	**0,98**	**0,96**	**1**	4,36	0,45	97,5
Eclat	**1246**	0,15	**0,99**	7,16	∞	**0,98**	**0,96**	**1**	4,36	0,45	97,5
Clearing	21,8	0,13	0,96	**9,5**	∞	0,94	0,84	0,98	2,06	**0,87**	99
NIGAR	9,8	**0,3**	0,98	5,51	∞	**0,98**	0,93	**1**	**2**	0,82	**100**
Coil2000											
Apriori	-	-	-	-	-	-	-	-	-	-	-
Eclat	-	-	-	-	-	-	-	-	-	-	-
Clearing	18,6	0,23	0,8	**5,93**	∞	0,69	0,63	0,85	2,08	0,75	99,81
NIGAR	**25,4**	**0,28**	**0,93**	3,65	∞	**0,89**	**0,86**	**0,98**	**2,04**	**0,82**	**99,96**
House16H											
Apriori	**1749917**	0,22	**0,97**	2,19	∞	0,83	0,45	0,76	8,65	0,28	**100**
Eclat	**1749917**	0,22	**0,97**	2,19	∞	0,83	0,45	0,76	8,65	0,37	**100**
Clearing	24,6	0,09	0,81	**13,12**	∞	0,74	0,69	0,55	2,09	0,72	83,41
NIGAR	6	**0,24**	0,92	3,67	15	**0,88**	**0,83**	**0,98**	**2**	**0,86**	82,18
Ionosphere											
Apriori	**211770656**	0,12	**0,98**	4,4	∞	**0,97**	0,72	**1**	10,79	0,03	**100**
Eclat	**211770656**	0,12	**0,98**	4,4	∞	**0,97**	0,72	**1**	10,79	0,28	**100**
Clearing	41	0,07	0,94	**55,85**	∞	0,91	**0,86**	0,98	**2,01**	**0,92**	92,03
NIGAR	24,8	**0,2**	0,85	3,77	∞	0,78	0,74	0,96	**2,01**	0,88	99,83
Letter											
Apriori	**3916**	0,14	0,88	4,66	∞	0,81	0,73	0,93	4,55	0,49	**99,49**
Eclat	**3916**	0,14	0,88	4,66	∞	0,81	0,73	0,93	4,55	0,49	**99,49**
Clearing	17,4	**0,16**	0,87	**7,28**	∞	0,77	0,72	0,93	2,11	0,81	98,26
NIGAR	18,8	0,15	**0,89**	5,94	∞	**0,84**	**0,76**	**0,96**	**2,01**	**0,87**	98,17
Magic											
Apriori	**9785**	0,19	**0,96**	2,73	∞	0,87	0,52	0,9	5,53	0,4	**99,96**
Eclat	**9785**	0,19	**0,96**	2,73	∞	0,87	0,52	0,9	5,53	0,4	**99,96**
Clearing	9,8	**0,34**	0,93	2,81	10,38	0,79	**0,85**	**0,99**	2,01	0,77	99,55
NIGAR	6,6	0,26	0,94	**3**	15,3	**0,91**	**0,85**	**0,99**	**2**	**0,85**	94,87
Movement Libras											
Apriori	-	-	-	-	-	-	-	-	-	-	-
Eclat	-	-	-	-	-	-	-	-	-	-	-
Clearing	23,6	0,24	0,94	**5,11**	∞	0,9	0,88	0,98	**2**	**0,87**	99,95
NIGAR	**27,8**	**0,27**	**0,98**	3,64	∞	**0,97**	**0,95**	**1**	**2**	0,86	**100**
Optdigits											
Apriori	-	-	-	-	-	-	-	-	-	-	-
Eclat	-	-	-	-	-	-	-	-	-	-	-
Clearing	17,8	0,22	0,84	**5,05**	∞	0,69	0,65	0,84	2,07	0,76	96,82
NIGAR	**28,8**	**0,28**	**0,87**	3,24	∞	**0,75**	**0,71**	**0,95**	**2,05**	**0,82**	**100**
Penbased											
Apriori	**1918**	0,14	**0,92**	**4,23**	∞	0,82	0,63	0,87	4,07	0,5	**99,5**
Eclat	**1918**	0,14	**0,92**	**4,23**	∞	0,82	0,63	0,87	4,07	0,5	**99,5**
Clearing	16,8	0,15	0,82	4,22	∞	0,73	0,63	0,87	**2,06**	0,84	98,08
NIGAR	15	**0,18**	0,89	3,64	∞	**0,83**	**0,73**	**0,95**	**2,06**	**0,88**	98,64
Pollution											
Apriori	**41510**	**0,13**	0,95	5,84	∞	0,93	0,86	0,98	5,88	0,38	**100**
Eclat	**41510**	**0,13**	0,95	5,84	∞	0,93	0,86	0,98	5,88	0,38	**100**
Clearing	48,6	0,05	**0,98**	**18,68**	∞	**0,97**	**0,9**	0,99	**2,01**	**0,96**	90,34
NIGAR	57,8	0,08	**0,98**	8,35	∞	**0,97**	0,84	**1**	2,07	0,93	**100**
Quake											
Apriori	18	0,25	**0,91**	1	1,15	0,11	-0,01	0,02	2,56	0,45	**90,55**
Eclat	18	0,25	**0,91**	1	1,15	0,11	-0,01	0,02	2,56	0,45	**90,55**
Clearing	**33,2**	0,03	0,86	**38,81**	∞	**0,84**	**0,76**	0,84	**2**	**0,88**	75,42
NIGAR	3,6	**0,28**	0,9	2,51	11,4	0,81	0,72	**0,95**	**2**	0,75	83,15
Satimage											
Apriori	-	-	-	-	-	-	-	-	-	-	-
Eclat	-	-	-	-	-	-	-	-	-	-	-
Clearing	15,2	**0,25**	**0,92**	**5**	∞	0,86	0,8	0,98	2,01	**0,82**	**96,34**
NIGAR	**24**	0,24	**0,92**	3,83	12,27	**0,88**	**0,85**	**0,99**	**2**	**0,82**	93,94

Tabla C.19: Resultados para las BDs Segment, Sonar, Spambase, Stock, Stulong, Texture, Wine, Wdbc y Vowel en la comparación entre los algoritmos clásicos y NIGAR

Algoritmos	*#R*	Med_{Sop}	Med_{Conf}	Med_{Lift}	Med_{Conv}	Med_{FC}	$Med_{Netconf}$	$Med_{Yule'sQ}$	Med_{Amp}	Med_{Div}	*%Ejem*
Segment											
Apriori	**3253152**	0,14	0,98	2,99	∞	0,85	0,48	0,84	8,57	0,33	**100**
Eclat	**3253152**	0,14	0,98	2,99	∞	0,85	0,48	0,84	8,57	0,41	**100**
Clearing	11,8	0,2	0,93	4,41	∞	0,91	0,88	0,94	2,02	0,86	98,09
NIGAR	12,6	**0,22**	**0,99**	**4,91**	∞	**0,98**	**0,97**	**1**	**2**	**0,88**	99,76
Sonar											
Apriori	-	-	-	-	-	-	-	-	-	-	-
Eclat	-	-	-	-	-	-	-	-	-	-	-
Clearing	**51,4**	0,06	**0,96**	**57,68**	∞	**0,93**	**0,91**	**0,99**	**2**	**0,94**	87,02
NIGAR	22,8	**0,26**	0,89	7,56	∞	0,81	0,77	0,97	2,01	0,84	**100**
Spambase											
Apriori	-	-	-	-	-	-	-	-	-	-	-
Eclat	-	-	-	-	-	-	-	-	-	-	-
Clearing	**32,4**	0,09	0,89	**115,83**	∞	0,83	0,81	0,88	2,03	**0,84**	**85,01**
NIGAR	11,2	**0,12**	**0,95**	73,02	∞	**0,9**	**0,87**	**0,98**	**2**	**0,84**	76,75
Spectfheart											
Apriori	-	-	-	-	-	-	-	-	-	-	-
Eclat	-	-	-	-	-	-	-	-	-	-	-
Clearing	39	0,12	0,97	**61,2**	∞	0,91	**0,94**	0,99	**2,01**	0,86	86,22
NIGAR	**72,2**	**0,13**	**0,98**	29,01	∞	**0,92**	0,92	**1**	2,03	**0,89**	**100**
Stock											
Apriori	**855**	0,13	0,91	4,77	∞	0,88	0,76	0,96	4,16	0,44	**99,48**
Eclat	**855**	0,13	0,91	4,77	∞	0,88	0,76	0,96	4,16	0,44	**99,48**
Clearing	14,6	0,21	0,91	**16,56**	∞	0,86	0,78	0,96	**2,06**	0,84	99,41
NIGAR	9,2	**0,28**	**0,95**	3,01	∞	**0,93**	**0,87**	**0,99**	2,07	**0,86**	99,41
Stulong											
Apriori	**89**	**0,31**	**0,93**	1,22	∞	0,43	0,14	0,29	3,26	0,43	**99,86**
Eclat	**89**	**0,31**	**0,93**	1,22	∞	0,43	0,14	0,29	3,26	0,43	**99,86**
Clearing	10,6	0,23	0,87	**29,51**	∞	**0,68**	**0,75**	**0,94**	2,1	0,72	97,2
NIGAR	4,2	0,26	0,79	16,82	∞	0,62	0,64	0,9	**2,04**	**0,78**	84,99
Texture											
Apriori	-	-	-	-	-	-	-	-	-	-	-
Eclat	-	-	-	-	-	-	-	-	-	-	-
Clearing	15,2	**0,29**	0,93	3,14	19,47	0,89	0,86	0,98	2,01	**0,83**	**97,26**
NIGAR	**17,4**	0,27	**0,97**	**3,72**	∞	**0,95**	**0,92**	**1**	**2**	0,82	95,85
Thyroid											
Apriori	-	-	-	-	-	-	-	-	-	-	-
Eclat	-	-	-	-	-	-	-	-	-	-	-
Clearing	**27**	0,11	0,87	**16,05**	∞	0,76	0,77	0,81	2,48	0,73	72,81
NIGAR	6,2	**0,23**	**0,93**	14,29	∞	**0,8**	**0,85**	**0,95**	**2,02**	**0,79**	**98,16**
Wdbc											
Apriori	-	-	-	-	-	-	-	-	-	-	-
Eclat	-	-	-	-	-	-	-	-	-	-	-
Clearing	**18,8**	0,19	0,92	**48,87**	∞	0,88	0,84	0,97	**2**	**0,86**	**99,51**
NIGAR	13,6	**0,27**	**0,96**	3,49	∞	**0,93**	**0,9**	**0,99**	**2**	0,84	98,53
Vehicle											
Apriori	**141107**	0,14	0,95	4,18	∞	0,91	0,76	0,95	6,4	0,47	**100**
Eclat	**141107**	0,14	0,95	4,18	∞	0,91	0,76	0,95	6,4	0,47	**100**
Clearing	13	0,26	0,96	**5,21**	∞	0,93	0,89	0,97	**2**	0,84	99,95
NIGAR	12,8	**0,27**	**0,97**	3,46	∞	**0,95**	**0,92**	**1**	**2**	**0,86**	99,98
Wine											
Apriori	**1348**	0,13	0,91	5,97	∞	0,87	0,82	0,97	4,07	0,68	**100**
Eclat	**1348**	0,13	0,91	5,97	∞	0,87	0,82	0,97	4,07	0,68	**100**
Clearing	23,2	0,14	**0,94**	**19,43**	∞	**0,91**	0,83	0,97	2,02	**0,92**	93,49
NIGAR	6,6	**0,28**	**0,94**	2,79	∞	0,9	**0,84**	**0,99**	**2**	0,86	94,61
Vowel											
Apriori	**235**	0,14	**0,98**	2,97	∞	**0,96**	0,69	**0,97**	3,48	0,7	**100**
Eclat	**235**	0,14	**0,98**	2,97	∞	**0,96**	0,69	**0,97**	3,48	0,7	**100**
Clearing	51,6	0,23	0,96	**92,68**	∞	0,78	**0,91**	0,5	**2,02**	0,81	92,1
NIGAR	8,4	**0,27**	0,95	5,81	∞	0,92	0,89	0,95	**2,02**	**0,85**	**100**

Bibliografía

[AA06] Alatas B. y Akin E. (2006) An efficient genetic algorithm for automated mining of both positive and negative quantitative association rules. *Soft Computing* 10(3): 230–237.

[AA08] Alatas B. y Akin E. (2008) MODENAR: Multi-objective diferential evolution algorithm for mining numeric association rules. *Applied Soft Computing* 8(1): 646–646.

[AFFL+11] Alcala-Fdez J., Fernandez A., Luego J., Derrac J., Garcia S., Sanchez L. y Herrera F. (2011) Keel data-mining software tool: Data set repository, integration of algorithms and experimental analysis framework. *Journal of Multiple-Valued Logic and Soft Computing* 17(2-3): 255–287.

[AFFPBH10] Alcala-Fdez J., Flugy-Pape N., Bonarini A. y Herrera F. (2010) Analysis of the effectiveness of the genetic algorithms based on extraction of association rules. *Fundamenta Informaticae* 98(1): 1–14.

[AFSG+09] Alcala-Fdez J., Sanchez L., Garcia S., del Jesus M., Ventura S., J. Garrell J. O., Romero C., Bacardit J., Rivas V., Fernandez J. y Herrera F. (2009) Keel: A software tool to assess evolutionary algorithms to data mining problems. *Soft Computing* 13(3): 307–318.

[AIS93] Agrawal R., Imielinski T. y Swami A. (1993) Mining association rules between sets of items in large databases. En *SIGMOD*, páginas 207–216. Washington D.C.

[AK04] Ahn K.-I. y Kim J.-Y. (2004) Efficient mining of frequent itemsets and a measure of interest for association rule mining. *Journal of Information & Knowledge Management* 3(3): 245–257.

[AS94] Agrawal R. y Srikant R. (1994) Fast algorithms for mining association rules. En *International Conference Large Data Bases*, páginas 487–499. Santiago de Chile, Chile.

[Bay98] Bayardo Jr. R. J. (1998) Efficiently mining long patterns from databases. En *Proceedings of the 1998 ACM SIGMOD International Conference on Management of Data*, SIGMOD '98, páginas 85–93. ACM, New York, NY, USA.

[BBSV02] Berzal F., Blanco I., Sanchez D. y Vila M. (2002) Measuring the accuracy and interest of association rules: A new framework. *Intelligent Data Analysis* 6(3): 221–235.

[BJQ+14] Bong K. K., Joest M., Quix C., Anwar T. y Manickam S. (January 2014) Selection and aggregation of interestigness measures: A review. *Journal of Theoretical and Applied Information Technology* 59(1): 146–166.

[BMS97] Brin S., Motwani R. y Silverstein C. (1997) Beyond market baskets: Generalizing association rules to correlations. En *ACM SIGMOD Conf*, páginas 265–276.

[BMUT97] Brin S., Motwani R., Ullman J. y Tsur S. (1997) Dynamic itemset counting and implication rules for market basket data. *ACM SIGMOD Record* 26(2): 255–264.

[Bor03] Borgelt C. (2003) Efficient implementations of Apriori and Eclat. En *Workshop on Frequent Itemset Mining Implementations*, volumen 90, páginas 280–296. CEUR Workshop Proc., Florida, USA.

[BPT+00] Bastide Y., Pasquier N., Taouil R., Stumme G. y Lakhal L. (2000) Mining minimal non-redundant association rules using frequent closed itemsets. En Lloyd J., Dahl V., Furbach U., Kerber M., Lau K.-K., Palamidessi C., Pereira L., Sagiv Y. y Stuckey P. (Eds.) *Proceedings of the First International Conference on Computational Logic*, volumen 1861 of *Lecture Notes in Computer Science*, páginas 972–986. Springer Berlin Heidelberg.

[Cav70] Cavicchio D. (1970) *Adapting search using simulated evolution.* PhD thesis, Univ. Michigan, Ann Arbor.

[CC09] Chen Q. y Chen Y. (2009) Discovery of structural and functional features in RNA pseudoknots. *IEEE Transactions on Knowledge and Data Engineering* 21(7): 974–984.

[CKOS91] Christopher K. Oei D. E. G. y Shang S.-J. (1991) Tournament selection, niching, and the preservation of diversity. Technical report, University of Illinois at Urbana-Champaign.

[CLV02] Coello C., Lamont G. y Veldhuizen D. V. (2002) *Evolutionary Algorithms for solving multi-objective problems.* Kluwer Academic Publishers.

[CMFLI06] Carlos M. Fonseca L. P. y Lopez-Ibanez M. (July 2006) An improved dimension-sweep al-gorithm for the hypervolume indicator. páginas 1157–1163. Vancouver, Canada.

[CRLA11] Chattopadhyay S., Rakesh S., Land L. y Acharya U. (2011) Studying infant mortality rate: A data mining approach. *Health and Technology* 1(1): 25–34.

[CV97] Cedeno W. y Vemuri V. R. (1997) On the use of niching for dynamic landscapes. En *IEEE Int. Conf. on Evolutionary Computation (ICEC'97)*, páginas 361–366. IEEE Publishing.

[DAPM02] Deb K., Agrawal S., Pratab A. y Meyarivan T. (2002) A fast and elitist multiobjective genetic algorithm: NSGA-II. *IEEE Transactions Evolutionary Computation* 6(2): 182–197.

[Deb01] Deb K. (2001) *Multi-objective optimization using evolutionary algorithms.* Kluwer Academic, EE.UU.

[Dem06] Demsar J. (2006) Statistical comparisons of classifiers over multiple data sets. *Journal of Machine Learning Research* 7: 1–30.

[DJ75] De Jong K. A. (1975) *An Analysis of the Behavior of a Class of Genetic Adaptive Systems.* PhD thesis, Ann Arbor, MI, USA.

[dJBC10] da Jiménez A., Berzal F. y Cubero J.-C. (Noviembre 2010) Frequent tree pattern mining: A survey. *Intelligent Data Analysis* 14(6): 603–622.

[dJGGP11] del Jesus M. J., Gomez J. A., Gonzalez P. y Puerta J. M. (2011) On the discovery of association rules by means of evolutionary algorithms. *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery* 1(5): 397–415.

[DL98] Dong G. y Li J. (1998) Interestingness of discovered association rules in terms of neighborhood-based unexpectedness. En *Proceedings of Second Pacific Asia Conference on Knowledge Discovery in Databases (PAKDD98)*, páginas 72–86. Melbourne.

[DMQS11] Das S., Maity S., Qu B.-Y. y Suganthan P. N. (2011) Real-parameter evolutionary multimodal optimization - a survey of the state-of-the-art. *Swarm and Evolutionary Computation* 1(2): 71–88.

[DMSaV03] Delgado M., Marín N., Sánchez D. y amparo Vila M. (2003) Fuzzy association rules: general model and applications. *IEEE Transactions on Fuzzy Systems* 11: 214–225.

[DPS05] Dubois D., Prade H. y Sudkamp T. (2005) On the representation, measurement, and discovery of fuzzy associations. *IEEE T. Fuzzy Systems* 13(2): 250–262.

[ES93] Eshelman L. J. y Schaffer J. D. (1993) Real-coded genetic algorithms and interval-schemata. En *Foundations of Genetic Algorithms*, volumen 2, páginas 187–202. Morgan Kauffman, San Mateo, CA, EE.UU.

[ES03] Eiben A. y Smith J. (2003) *Introduction to Evolutionary Computing*. SpringerVerlag.

[EZdF03] E. Zitzler L. Thiele M. L. C. M. F. y da Fonseca V. G. (2003) Performance assessment of multiobjective optimizers: An analysis and review. *IEEE Transactions on Evolutionary Computation* 7(2): 117–32.

[FF93] Fonseca C. y Fleming P. (1993) Genetic algorithms for multiobjective optimization: Formulation, discussion and generalization. En *5th International Conference on Genetic Algorithms*, páginas 416–423. San Mateo, CA.

[Fin93] Finner H. (1993) On a monotonicity problem in step-down multiple test procedures. *Journal of the American Statistical Association* 88(423): 920–923.

[FLF00] Fidelis M., Lopes H. y Freitas A. (2000) Discovering comprehensible classification rules with a genetic algorithm. En *Proceedings of the 2000 Congress on Evolutionary Computation*, páginas 805–810. CA, USA.

[FMMT96] Fukuda T., Morimoto Y., Morishita S. y Tokuyama T. (1996) Mining optimized association rules for numeric attributes. En *Proceedings of the Fifteenth ACM SIGACT-SIGMOD-SIGART Symposium on Principles of Database Systems*, PODS '96, páginas 182–191. ACM, New York, NY, USA.

[Fri37] Friedman M. (1937) The use of ranks to avoid the assumption of normality implicit in the analysis of variance. *Journal of the American Statistical Association* 32(200): 675–701.

[GFLH09] Garcia S., Fernandez A., Luengo J. y Herrera F. (2009) A study of statistical techniques and performance measures for genetics-based machine learning: Accuracy and interpretability. *Soft Computing* 13(10): 959–977.

[GG13] Glibovets N. N. y Gulayeva N. M. (2013) A review of niching genetic algorithms for multimodal function optimization. *Cybernetics and Systems Analysis* 49(6): 815–820.

[GH06] Geng L. y Hamilton H. (2006) Interestingness measures for data mining: A survey. *ACM Computing Surveys* 38(3).

[GH07] Guillet F. y Hamilton H. J. (Eds.) (2007) *Quality Measures in Data Mining*, volumen 43 of *Studies in Computational Intelligence*. Springer.

[GH08] Garcia S. y Herrera F. (2008) An extension on statistical comparisons of classifiers over multiple data sets for all pairwise comparisons. *Journal of Machine Learning Research* 9: 2579–2596.

[GJ05] Ghosh A. y Jain L. (Eds.) (2005) *Evolutionary computation in data mining*. Number 163 in Studies in fuzziness and soft computing. Springer, Berlin [u.a.].

[Gla13] Glass D. H. (Mayo 2013) Confirmation measures of association rule interestingness. *Knowledge-Based Systems* 44: 65–77.

[GMLH09] Garcia S., Molina D., Lozano M. y Herrera F. (2009) A study on the use of non-parametric tests for analyzing the evolutionary algorithms' behaviour: A case study on the cec'2005 special session on real parameteroptimization. *Journal of Heuristics* 15: 617–644.

[GN04] Ghosh A. y Nath B. (2004) Multi-objective rule mining using genetic algorithms. *Information Sciences* 163(1-3): 123–133.

[Goe02] Goethals B. (2002) *Efficient Frequent Pattern Mining*. PhD thesis, University of Limburg, Belgium.

[Gol89] Goldberg D. (1989) *Genetic Algorithms in Search, Optimization and Machine Learning*. Addison-Wesley Longman Publishing Co., Inc.

[GR87] Goldberg D. E. y Richardson J. (1987) Genetic algorithms with sharing for multimodal function optimization. En *Proceedings of the Second International Conference on Genetic Algorithms on Genetic Algorithms and Their Application*, páginas 41–49. L. Erlbaum Associates Inc., Hillsdale, NJ, USA.

[Har94] Harik G. (1994) Finding multiple solutions in problems of bounded difficulty. IlliGAL report 94002, Illinois Genetic Algorithms Laboratory, University of Illinois at Urbana-Champaign.

[HCGdJ10] Herrera F., Carmona C., González P. y del Jesus M. (2010) An overview on subgroup discovery: foundations and applications. *Knowledge and Information Systems* páginas 1–31.

[HK06] Han J. y Kamber M. (2006) *Data Mining: Concepts and Techniques*. Morgan Kaufmann, second edition edition.

[HNG94] Horn J., Nafpliotis N. y Goldberg D. (1994) A niched pareto genetic algorithm for multiobjective optimization. En *First IEEE Conf. on Evolutionary Computation, IEEE World Congress on Computational Intelligence*, páginas 82–87. Piscataway, NJ.

[Hol79] Holm S. (1979) A simple sequentially rejective multiple test procedure. *Scandinavian Journal of Statistics* 6: 65–70.

[HPYM04] Han J., Pei J., Yin Y. y Mao R. (2004) Mining frequent patterns without candidate generation: A frequent-pattern tree approach. *Data Mining and Knowledge Discovery* 8(1): 53–87.

[HR77] Holland J. H. y Reitman J. S. (Junio 1977) Cognitive systems based on adaptive algorithms. *SIGART Bull.* (63): 49–49.

[ID80] Iman R. y Davenport J. (1980) Approximations of the critical region of the friedman statistic. *Communications in Statistics - Theory and Methods* 9(6): 571–595.

[JBVW06] Janssens D., Brijs T., Vanhoof K. y Wets G. (2006) Evaluating the performance of cost-based discretization versus entropy and error-based discretization. *Computers and Operations Research* 33(11): 3107–3123.

[KCJL02] Kim J., Cho D., Jung H. y Lee C. (2002) Niching genetic algorithm adopting restricted competition selection combined with pattern search method. *IEEE Transactions on magnetic* 38(2): 1001 – 1004.

[Klo96] Klosgen W. (1996) Explora: A multipattern and multistrategy discovery assistant. En *Advances in Knowledge Discovery and Data Mining*, páginas 249–271. AAAI.

[LBPC02] Li J. P., Balazs M. E., Parks G. T. y Clarkson P. J. (2002) A species conserving genetic algorithm for multimodal function optimization. *Evolutionary Computation* 10(3): 207–234.

[LCJ99] Lee C., Cho D. y Jung H. (1999) Niching genetic algorithm with restricted competition selection for multimodal function optimization. *IEEE transactions on magnetics* 35(3): 1722 – 1725.

[Lee07] Lee C.-H. (2007) A hellinger-based discretization method for numerica attributes in classification learning. *Knowledge-Based Systems* 20(4): 419–425.

[LHKM04] Lozano M., Herrera F., Krasnogor N. y Molina D. (Septiembre 2004) Real-coded memetic algorithms with crossover hill-climbing. *Evol. Comput.* 12(3): 273–302.

[LHM98] Liu B., Hsu W. y Ma Y. (August 1998) Integrating classification and association rule mining. En *Proceedings of the 4th international conference on Knowledge Discovery and Data mining (KDD'98)*, páginas 80–86. AAAI Press.

[LHTD02] Liu H., Hussain F., Tan C. y Dash M. (2002) Discretization: An enabling technique. *DataMining and Knowledge Discovery* 6(4): 393–423.

[Li05] Li X. (2005) Efficient differential evolution using speciation for multimodal function optimization. En *Proceedings of the 2005 Conference on Genetic and Evolutionary Computation*, GECCO '05, páginas 873–880. ACM, New York, NY, USA.

[LW02] Lin C.-Y. y Wu W.-H. (2002) Niche identification techniques in multimodal genetic search with sharing scheme. *Advances in Engineering Software* 33(1112): 779 – 791.

[LW09] Li J.-P. y Wood A. S. (2009) An adaptive species conservation genetic algorithm for multimodal optimization. *International Journal for Numerical Methods in Engineering* 79(13): 1633–1661.

[LZ09] Li H. y Zhang Q. (2009) Multiobjective optimization problems with complicated pareto sets, MOEA/D and NSGA-II. *IEEE Transactions on Evolutionary Computation* 13(2): 284–302.

[Mah92] Mahfoud S. W. (1992) Crowding and preselection revisited. *Parallel Problem Solving from Nature* páginas 27–36.

[Mah95] Mahfoud S. (1995) *Niching method for genetic algorithms*. PhD thesis, Doctoral Dissertation, Technical Report, Department of Computer Science, University of Illinois at Urbana-Champaign, Urbana, IL, USA, Illinois Genetic Algorithms Laboratory, IlliGAL.

[MAR01] Mata J., Alvarez J. y Riquelme J. (April 2001) Mining numeric association rules with genetic algorithms. En *5th International Conference on Artificial Neural Networks and Genetic Algorithms*. Taipei, Taiwan.

[MAR02] Mata J., Alvarez J. y Riquelme J. (March 2002) An evolutionary algorithm to discover numeric association rules. En *ACM Symposium on Applied Computing*. Madrid, Spain.

[MBMATR11] Martinez-Ballesteros M., Martinez-Alvarez F., Troncoso A. y Riquelme J. C. (Octubre 2011) An evolutionary algorithm to discover quantitative association rules in multidimensional time series. *Soft Comput.* 15(10): 2065–2084.

[MBMATR14] Martinez-Ballesteros M., Martinez-Alvarez F., Troncoso A. y Riquelme J. (2014) Selecting the best measures to discover quantitative association rules. *Neurocomputing* 126(0): 3 – 14.

[Mer13] Mersmann O. (2013) *emoa: Evolutionary Multiobjective Optimization Algorithms*. http://cran.r-project.org/web/packages/emoa/.

[MG99] Mengshoel O. J. y Goldberg D. E. (13-17 July 1999) Probabilistic crowding: Deterministic crowding with probabilisitic replacement. En *Proceedings of the Genetic and Evolutionary Computation Conference*, volumen 1, páginas 409–416. Morgan Kaufmann, Orlando, Florida, USA.

[Mic96] Michalewicz Z. (1996) *Genetic algorithms + data structures = evolution programs*. Springer-Verlag.

[Mie99] Miettinen K. (1999) *Nonlinear Multiobjective Optimization*. Kluwer, Norwell, MA.

[MMBC14] Mukhopadhyay A., Maulik U., Bandyopadhyay S. y Coello C. A. C. (2014) A survey of multiobjective evolutionary algorithms for data mining: Part II. *IEEE Transactions Evolutionary Computation* 18(1): 20–35.

[PAMV12] Pachón Álvarez V. y Mata Vázquez J. (Enero 2012) An evolutionary algorithm to discover quantitative association rules from huge databases without the need for an a priori discretization. *Expert Syst. Appl.* 39(1): 585–593.

[PBTL99] Pasquier N., Bastide Y., Taouil R. y Lakhal L. (1999) Discovering frequent closed itemsets for association rules. En *Proceedings of the 7th International Conference on Database Theory*, ICDT '99, páginas 398–416. Springer-Verlag, London, UK, UK.

[Pét97] Pétrowski A. (1997) A new selection operator dedicated to speciation. En *7th international conference on genetic algorithms (ICGA)*, páginas 144–151. San Mateo. USA.

[PL06] Parrott D. y Li X. (Aug 2006) Locating and tracking multiple dynamic optima by a particle swarm model using speciation. *IEEE Transactions on Evolutionary Computation* 10(4): 440–458.

[PPH12] Pérez E., Posada M. y Herrera F. (Junio 2012) Analysis of new niching genetic algorithms for finding multiple solutions in the job shop scheduling. *J. Intell. Manuf.* 23(3): 341–356.

[PS91] Piatetsky-Shapiro G. (1991) Discovery, analysis and presentation of strong rules. En MIT Press C. (Ed.) *Knowledge Discovery in Databases*, páginas 229–248. Cambridge, MA.

[PSC11] Palacios A. M., Sánchez L. y Couso I. (2011) Future performance modeling in athletism with low quality data-based genetic fuzzy systems. *Multiple-Valued Logic and Soft Computing* 17(2-3): 207–228.

[Pét96] Pétrowski A. (1996) A clearing procedure as a niching method for genetic algorithms. En *International Conference on Evolutionary Computation*, páginas 798–803. Japan.

[PT00] Padmanabhan B. y Tuzhilin A. (2000) Small is beautiful: Discovering the minimal set of unexpected patterns. En *Proceedings of the Sixth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '00, páginas 54–63. ACM, New York, NY, USA.

[QNMB11] Qodmanan H., Nasiri M. y Minaei-Bidgoli B. (2011) Multi objective association rule mining with genetic algorithm without specifying minimum support and minimum confidence. *Expert Systems with Applications* 38: 288–298.

[RBV03] Renesse R., Birman K. y Vogels W. (2003) Astrolabe: A robust and scalable technology for distributed system monitoring, management, and data mining. *ACM Transactions on Computer Systems* 21(2): 164–206.

[RMS98] Ramaswamy S., Mahajan S. y Silberschatz A. (1998) On the discovery of interesting patterns in association rules. En *24rd International Conference on Very Large Data Bases*, páginas 368–379. San Francisco, CA, USA.

[Ros67] Rosenberg R. (1967) Simulation of genetic populations with biochemical properties. Master's thesis, Univ. Michigan, Ann Harbor, Michigan.

[SA96] Srikant R. y Agrawal R. (1996) Mining quantitative association rules in large relational tables. En *ACM SIGMOD International Conference on Management of data (SIGMOD96)*, páginas 1–12. Montreal, Quebec, Canada.

[Sat12] Sathi A. (2012) *Big Data Analytics: Disruptive Technologies for Changing the Game*. MC Press.

[SAVN07] Salleb-Aouissi A., Vrain C. y Nortet C. (2007) Quantminer: A genetic algorithm for mining quantitative association rules. En *Proceedings of the 20th International Joint Conference on Artifical Intelligence*, IJCAI'07, páginas 1035–1040. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA.

[SB75] Shortliffe E. y Buchanan B. (1975) A model of inexact reasoning in medicine. *Mathematical Biosciences* 23: 351–379.

[SBM98] Silverstein C., Brin S. y Motwani R. (1998) Beyond market baskets: Generalizing association rules to dependence rules. *Data Mining and Knowledge Discovery* 2(1): 39–68.

[Sch85] Schaffer J. (1985) Multiple objective optimization with vector evaluated genetic algorithms. En *First International Conference on Genetic Algorithms*, páginas 93–100. Pittsburgh, USA.

[SD94] Srinivas N. y Deb K. (1994) Multiobjective optimization using nondominated sorting in genetic algorithms. *Evolutionary Computation* 2(3): 221–248.

[She03] Sheskin D. (2003) *Handbook of Parametric and Nonparametric Statistical Procedures*. Chapman & Hall/CRC, London, U.K./Boca Raton.

[SK98] Sareni B. y Krähenbühl L. (1998) Fitness Sharing and Niching Methods Revisited. *IEEE Transactions on Evolutionary Computation* 2(3): 97–106.

[Smi80] Smith S. F. (1980) *A Learning System Based on Genetic Adaptive Algorithms*. PhD thesis, Pittsburgh, PA, USA. AAI8112638.

[SR11] Srinivasan S. y Ramakrishnan S. (2011) Evolutionary multi objective optimization for rule mining: a review. *Artificial Intelligence Review* 36(3): 205–248.

[THA07] THABTAH F. (Marzo 2007) A review of associative classification mining. *Knowl. Eng. Rev.* 22(1): 37–65.

[TKS02] Tan P., Kumar V. y Srivastava J. (2002) Selecting the right interestingness measure for association patterns. En *Proceedings of the 8th International Conference on Knowledge Discovery and Data Mining (KDD 2002)*, páginas 32–41. Edmonton, Canada.

[TLY08] Tsai C., Lee C. y Yang W. (2008) A discretization algorithm based on class-attribute contingency coefficient. *Information Science* 178(3): 714–731.

[TM05] Tang Z. y MacLennan J. (2005) *Data Mining With SQL Server 2005*. Wiley, Indianapolis, IN.

[TRD12] Taniar D., Rahayu W. y Daly O. (2012) Mining hierarchical negative association rules. *International Journal of Computational Intelligence Systems* 5(3): 434–451.

[VOOS09] Villar J., Otero A., Otero J. y Sánchez L. (2009) Taximeter verification using imprecise data from gps. *Eng. Appl. of AI* 22(2): 250–260.

[WFH11] Witten I. H., Frank E. y Hall M. A. (2011) *Data Mining: Practical Machine Learning Tools and Techniques*. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 3rd edition.

[Wil45] Wilcoxon F. (1945) Individual comparisons by ranking methods. *Biometrics* 1(6): 80–83.

[WZZ04] Wu X., Zhang C. y Zhang S. (2004) Efficient mining of both positive and negative association rules. *ACM Transactions on Information Systems* 22(3): 381–405.

[YG93] Yin X. y Germay N. (1993) A fast genetic algorithm with sharing scheme using cluster analysis methods in multi-modal function optimization. En *Proceedings of the International Conference on Artificial Neural Nets and Genetic Algorithms*, number 450-457.

[YZZ09] Yan X., Zhang C. y Zhang S. (2009) Genetic algorithm-based strategy for identifying association rules without specifying actual minimum support. *Expert Systems with Applications* 36(2): 3066–3076.

[Zak00] Zaki M. (2000) Scalable algorithms for association mining. *IEEE Transactions on Knowledge and Data Engineering* 12(3): 372–390.

[ZGBD09] Zhu X., Goldberg A. B., Brachman R. y Dietterich T. (2009) *Introduction to Semi-Supervised Learning*. Morgan and Claypool Publishers.

[ZL07] Zhang Q. y Li H. (2007) Moea/d: A multiobjective evolutionary algorithm based on decomposition. *IEEE Transactions on Evolutionary Computation* 11(6): 712–731.

[ZLT01] Zitzler E., Laumanns M. y Thiele L. (2001) Spea2: Improving the strength pareto evolutionary algorithm for multiobjective optimization. En *Evolutionary Methods for Design, Optimization and Control: Applications to Industrial and Societal Problems*, páginas 95–100. Barcelona, Spain.

[ZZ02] Zhang C. y Zhang S. (2002) Association rule mining: Models and algorithms. En *Lecture Notes in Computer Science, LNAI 2307*.